Non-local Finite Element Model for Rigid Origami

by

Deepakshyam Krishnaraju

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved May 2014 by the
Graduate Supervisory Committee:

Hanqing Jiang, Chair
Hongyu Yu
Marc Mignolet

ARIZONA STATE UNIVERSITY

August 2014

ABSTRACT

Origami is an art transforming a flat sheet of paper into a sculpture. Among various types of origami, the focus is on a particular class called the 'Rigid Origami' ("RO"). A Rigid Origami, unlike other forms, is not intended to be folded into fancy shapes. On the contrary, an RO has a simple and a geometrically well-defined crease pattern and does not have curved/smudged faces. The folds can be carried out by a continuous motion in which, at each step, each face of the origami is completely flat. As a result, these planar faces experience very minimal strain due to loading. This property allows it to be used to fold surfaces made of rigid materials. Tapping into the geometrical properties of RO will open a new field of research with great practical utility.

Analyzing each new RO pattern will require generating numerous prototypes; this is practically impossible to do, as it consumes a lot of time and material. The advantages of Finite Element Analysis/numerical modeling become very clear in this scenario. A new design concept may be modeled to determine its real world behavior under various load environments and may, therefore, be refined prior to the creation of drawings, when changes are inexpensive.

Since an RO undergoes a non-local deformation when subjected to a disturbance, the usage of conventional FEA will not produce accurate results. A non-local element model was developed which can be used in conjunction with the finite element package ABAQUS, via its user-defined element (UEL). This model was tested on two RO patterns, namely Miura-Ori and Ron Resch, by carrying out basic simulations.

There are many other interesting origami patterns, exhibiting different meta-material properties, yet to be explored. This Finite Element Approach equips researchers with necessary tools to study those options in great detail.

DEDICATION

I dedicate my dissertation work to my family and many friends. A special feeling of gratitude goes to my loving parents, Mr. and Mrs. Krishanraju, whose words of encouragement took me through some tough times.

I also dedicate this dissertation to my friends who have supported me throughout the process. I will always appreciate all they have done: especially Cheng Lv and Rajesh Kumar for helping me develop my technology skills; Saisree Raman and Vivek Kopparthi for the many hours of proofreading; Ajit Raghunathan and Siva Kumar for helping me with formatting.

ACKNOWLEDGEMENTS

I wish to thank my committee members who were more than generous with their expertise and precious time. A special thanks to Dr. Hanqing Jiang, my committee chairman for his countless hours of reflecting, reading, encouraging, and, most of all, patience throughout the entire process. Thank you, Dr. Hongyu Yu and Dr. Marc Mignolet, for agreeing to serve on my committee.

I would like to acknowledge and thank SEMTE for allowing me to conduct my research and providing any assistance requested. Special thanks go to the members of the staff development and human resources department for their continued support. Finally, I would like to acknowledge the seed grant support from the Office of Associate Dean for Research at Ira A. Fulton School of Engineering, Arizona State University.

TABLE OF CONTENTS

Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 : INTRODUCTION

Origami is an art transforming flat sheets of paper into a structure with depth and definition. This ancient art originated in Japan in the late seventeenth century and became popular in mid-1900s. Since then, it has evolved into a modern art form, and recently people have started using origami in engineering and structural applications. There are many classes of origami. Even though factors of classification are not unanimous among many scholar groups, below mentioned categories are the ones widely accepted.

Action Origami not only covers static objects. Some of them can also be moving; some move in clever ways. It also includes origami that flies, requires inflation to complete, or uses the kinetic energy of the operator's hands, applied at a certain region of the model.

Modular Origami consists of putting a number of identical pieces together to form a complete model. Normally, the individual pieces are simple, but the final assembly may be tricky. This style originated from some Chinese refugees while they were detained in America and is also called Golden Venture folding, named for the ship they came on.

Wet Folding is an origami technique for producing models with gentle curves rather than geometric straight folds and flat surfaces. The paper is dampened so it can be molded easily; the final model keeps its shape when it dries. It can be used, for instance, to produce very natural looking animal models.

Pureland Origami is origami with the restriction that only one fold may be done at a time; more complex folds, like reverse folds, are not allowed; and all folds have straightforward locations. It was to help inexperienced folders or those with limited motor skills.

Kirigami is a Japanese term for paper cutting. Cutting was often used in traditional Japanese origami, but modern innovations in technique have made the use of cuts unnecessary. Most origami designers no longer consider models with cuts to be origami, instead using the term "kirigami" to describe them.

Origami Tessellations is a branch of origami that has grown in popularity recently. A tessellation is a collection of figures filling a plane, with no gaps or overlaps. They are called "rigid origami" because the planes are rigid; in other words, they experience negligible strain.

Rigid origami uses flat rigid sheets joined by hinges for folding structures. These hinges form the creases of the origami and also act as a mechanical linkage. These linkages support the load and minimize the strain exerted on the flat planes. The following figures show a couple of the famous rigid origami patterns (RO), namely, Miura-Ori (MO) and Ron Resch (RR).

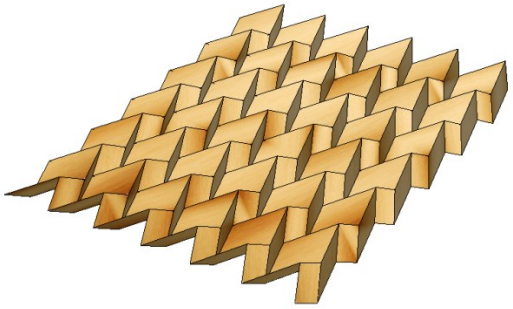


Figure 1.1: Miura-Ori folding

Figure 1.2: Ron Resch folding

Its construction begins with folding the creases into the paper. After all the creases are defined, the flat sheet of paper can be transformed into an origami by just pushing the vertices into suitable zones. By suitable zones, I refer to the set of all positions of the vertices, which defines the characteristic shape of origami. Once the basic shape is assumed, the origami has many intermediate configurations in these zones. This ability to make a smooth transition from one intermediate configuration to another was one of the main reasons for deploying them in many engineering applications.

A Miura folding (see Figure 1.1) was used to fold up a solar sail for ease of transportation, to later be deployed in outer space. A fully stretched solar sail is an area which would be impossible to transport without such folding mechanisms. This marked the beginning of the union between engineering and art and opened up a new field of research.

Since RO have bright prospects in engineering applications, much attention has been focused on developing a finite element program to accurately model them. This will act as an excellent tool for predicting the characteristics of RO with varying sizes. The biggest challenge was capturing the non-local nature of displacement exhibited by the material points in the origami. This will be discussed in great depth in the following sections.

CHAPTER 2 : DISPLACEMENT-BASED FINITE ELEMENT ANALYSIS

2.1 GENERAL IDEA

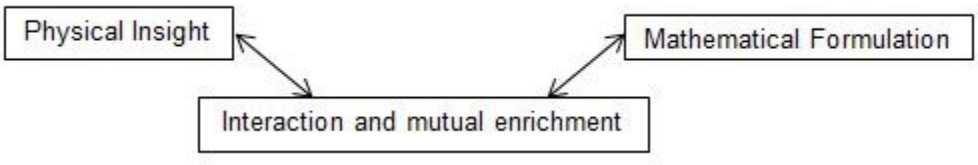


Figure 2.1: Basic Idea behind FEA

Finite element analysis can be considered a virtual experiment conducted in a computer to reproduce an actual test as accurately possible. It is computationally impossible to reproduce the actual environment without dumbing it down with some assumptions. These assumptions make the experiment feasible and generate results that are acceptable when compared with the actual test results. Since it saves a lot time, money, and material waste, it is being widely used in many fields like mining, automobile engineering, nuclear engineering and bio-medical engineering.

2.2 TYPES OF EQUILIBRIUM

There are primarily three types of equilibrium found in nature, namely: stable, unstable and neutral equilibrium. They are defined as follows.

1. Stable equilibrium: the kind of equilibrium of a body so placed that if disturbed, it returns to its former position. The center of gravity is below the point of suspension.

2. Unstable equilibrium: the kind of equilibrium in which the body, if disturbed, does not tend to return to its former position. The center of gravity is located above the point of suspension.

3. Neutral Equilibrium: the kind of equilibrium of a body so placed that, when moved slightly, it neither tends to return to its former position nor depart more widely from it . The center of gravity is exactly at the point of support.

It is very important to obtain the stable equilibrium configuration of the system under study before carrying out any analysis. Otherwise, the system will be constantly under the

influence of unknown external force, which makes it practically impossible to obtain a meaningful solution.

2.3 Steps in the development of a Finite Element Model

It is very hard to determine the configuration of the minimum energy of a system. So, a finite element model has to be developed in order to reach the equilibrium state of the system. Any finite element procedure contains the following steps.

1. Development of a model: Model development is the process of identifying the ingredients of a model which can provide the quantitative and qualitative predictions. The objective is find simplest model which can replicate the behavior of interest.

2. Formulation of the governing differential equations: A differential equation is a mathematical equation that relates some function of one or more variables with its derivatives. Differential equations arise whenever a deterministic relation, involving some continuously varying quantities (modeled by functions) and their rates of change in space and/or time (expressed as derivatives), is known or postulated. Every natural system has continuously varying quantities that correspond to various stimuli and can be represented by a system of differential equations.

3. Discretization of the equations: In mathematics, discretization concerns the process of transferring continuous models and equations into discrete counterparts. This process is usually carried out as a first step toward making them suitable for numerical evaluation and implementation on digital computers. Only the simplest differential equations are solvable by explicit formulae. For others, solutions are numerically approximated by using computer algorithms.

4. Solution of the equations: This is a very crucial step. Based on the type of input provided, it can either be used to obtain the equilibrium configuration or to carry out an actual analysis. An understanding of advantages and disadvantages and the approximate computer time required for various solution procedures are invaluable in the step. A poor judgment can lead to very long run-times and even unreasonable results.

5. Interpretation of the results: The user's role is most crucial in the interpretation of results. In addition to the approximations inherent even in linear finite element models, nonlinear analyses are often sensitive to many factors that can make a single simulation quite misleading. Nonlinear solids can undergo instabilities, their response can be sensitive to imperfections, and the results can depend dramatically on material parameters. Unless the user is aware of these phenomena, the possibility of a misinterpretation of simulation results is quite possible.

The above steps have been summarized in the form for an intuitive flow diagram, as shown in Figure 2.2:

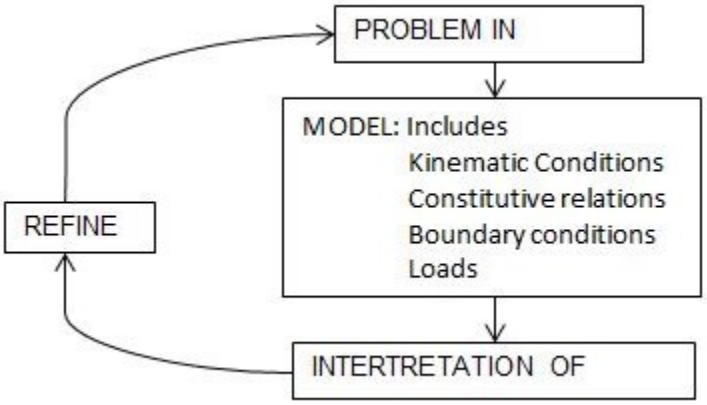


Figure 2.2: Steps in the development of FE Model

2.4 PRINCIPLE OF VIRTUAL WORK

The PoVW states that, for any variation satisfying the displacement boundary condition, the external virtual work is equal to the internal virtual work. In other words, the total work done by a force on a body equals the work done within the body due to movement of material particles, provided the enforced displacements constraints remain unchanged.

Time acts as a pseudo-variable in analyses with time independent material properties. It directly corresponds to the magnitude of applied load. It is an actual variable in dynamic analysis and analysis with time dependent material properties (Creep). Here, Δt must be chosen with respect to physics of the problem, cost of the simulation and numerical technique involved. At the end of each load step, it's important to satisfy three basic requirements: equilibrium, compatibility,

stress strain law. This is achieved – in an approximate manner, using finite elements – by the application of the Principle of Virtual Work.

2.4.1 Fundamental equation of Non-linear Finite Element Analysis

Consider a body in its initial configuration (in black), subjected to an arbitrary displacement boundary condition. This body gets deformed to an arbitrary shape state at time t (in red) as shown Figure 2.3. According to Principle of Virtual Work ("PoVW"), when a small arbitrary variation in displacement is applied to this configuration, the external virtual work corresponding to the virtual strain must be equal to the internal virtual work, provided the displacement boundary conditions are satisfied. This is true for all similar variations, even for all infinite possible configurations. An example of this variation is shown in blue, in Figure 2.4. But the focus is on only those configurations that can be captured using the interpolation functions.

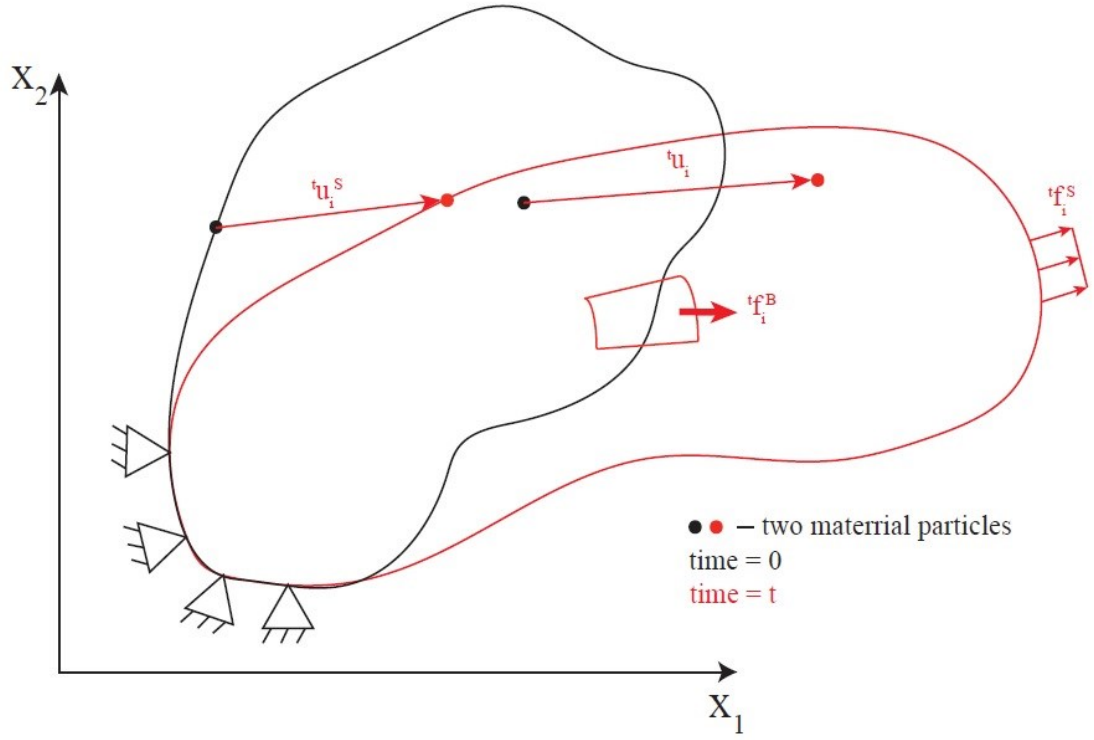


Figure 2.3: Illustration for the Principle of Virtual Work

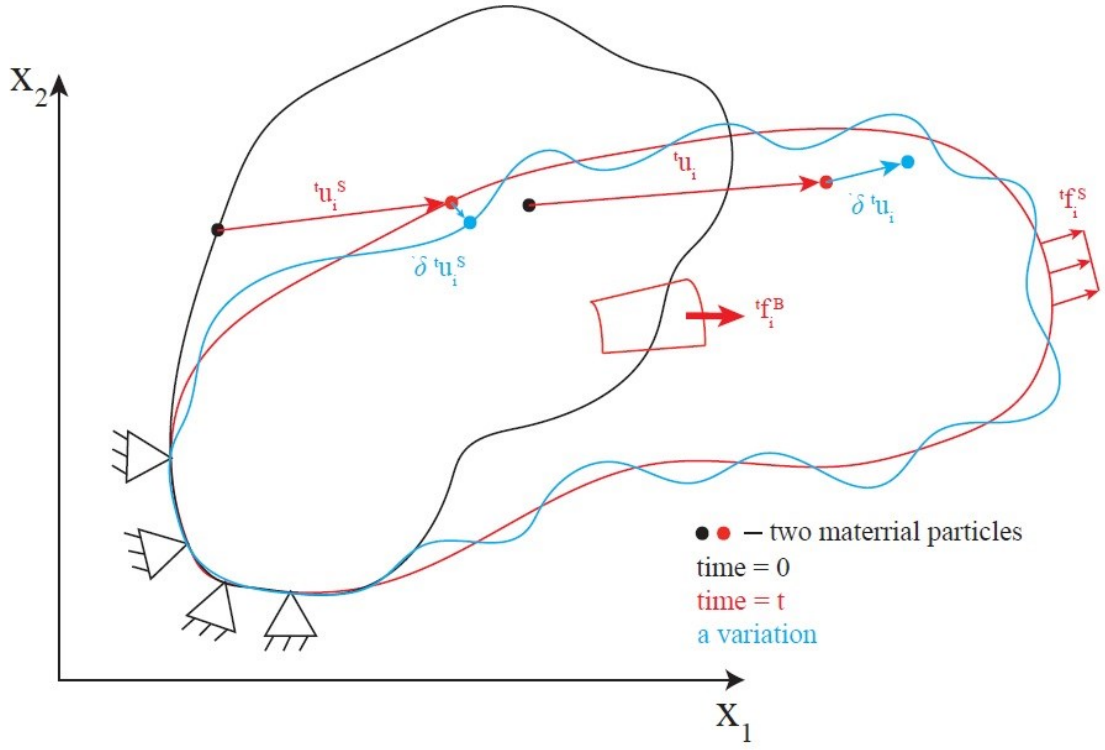


Figure 2.4: Displacement variation at time t

Here, ${}^{t}\mathbf{u}_i^S$ and ${}^{t}\mathbf{u}_i$ denote the displacement vectors of the material points at surface and inside the body respectively. ${}^{t}\mathbf{f}_i^B$, denotes the body force vector per unit volume, and ${}^{t}\mathbf{f}_i^S$ is the surface traction vector causing the change in configuration. The terms ${}^{t}\delta\mathbf{u}_i^S$ and ${}^{t}\delta\mathbf{u}_i$ denote the virtual displacement vectors of the material points.

The total external virtual work:

$$ {}^{t}\mathfrak{R} = \int_{{}^{t}V} {}^{t}\mathbf{f}_i^B \delta_t \mathbf{u}_{ij} \, {}^{t}dV + \int_{{}^{t}S} {}^{t}\mathbf{f}_i^S \delta_t \mathbf{u}_i \, {}^{t}dS . \tag{1} $$

The total internal virtual work:

$$ \int_{{}^{t}V} {}^{t}\boldsymbol{\tau}_{ij} \delta_t \mathbf{e}_{ij} \, {}^{t}dV = {}^{t}\mathfrak{R} . \tag{2} $$

According toPoVW, the equations (1) and (2) can be equated, provided the displacement boundary conditions are satisfied for arbitrary variations in displacement.

U_{tot} : Total Elastic Strain Energy in the element

k_i : Stiffness of dihederal angle along direction i

${}_i\alpha_j$: j^{th} dihederal angle along direction i

${}_i\alpha_{eq}$: Equilibrium dihederal angle along direction i

n_{TD} : Number of types of dihederal angles in the element

n_{TE} : Total number of dihederal angles in the element

Note: integrating the PoVW by parts yields the governing differential equations and stress boundary conditions

2.4.2 BASIC CONTINUUM MECHANICS

When a body gets deformed, it gets stretched or rotated or even both. The motion of the body is described by a function

$$x = \phi(X,t) \qquad X \in [X_a, X_b]. \tag{3}$$

This is called the map between the initial and the current/final configuration. The displacement is given by subtracting the current and initial coordinates

$$u = x - X \Rightarrow u = \phi(X,t) - X. \tag{4}$$

When the body deforms from its initial configuration, it undergoes both rotation and linear translation. A tensor called the deformation gradient (**F**) is defined such that it captures these motions.

$${}^t_0\mathbf{F} = {}^t_0\mathbf{X} = \left({}_0\nabla\, {}^t\mathbf{x}^T\right)^T \tag{5}$$

$$ {}_0\nabla = \begin{bmatrix} \dfrac{\partial}{\partial {}^0x_1} \\ \dfrac{\partial}{\partial {}^0x_2} \\ \dfrac{\partial}{\partial {}^0x_3} \end{bmatrix} \qquad {}^tx^T = \begin{bmatrix} {}^tx_1 & {}^tx_2 & {}^tx_3 \end{bmatrix}. \tag{6} $$

In other words,

$$ {}^t_0\mathbf{X} = \begin{bmatrix} \dfrac{\partial {}^tx_1}{\partial {}^0x_1} & \dfrac{\partial {}^tx_1}{\partial {}^0x_2} & \dfrac{\partial {}^tx_1}{\partial {}^0x_3} \\ \dfrac{\partial {}^tx_2}{\partial {}^0x_1} & \dfrac{\partial {}^tx_2}{\partial {}^0x_2} & \dfrac{\partial {}^tx_2}{\partial {}^0x_3} \\ \dfrac{\partial {}^tx_3}{\partial {}^0x_1} & \dfrac{\partial {}^tx_3}{\partial {}^0x_2} & \dfrac{\partial {}^tx_3}{\partial {}^0x_3} \end{bmatrix}. \tag{7} $$

2.4.3 LINEARIZATION OF PRINCIPLE OF VIRTUAL WORK

There is a large amount of non-linarites arising from kinematics (material relationships). This makes computation costly and requires complicated algorithms to determine the configurations at successive time steps. Hence, many higher order terms in the principle of virtual work are neglected, and this is referred to as linearization. The process can be represented by a flow diagram as shown in Figure 2.5

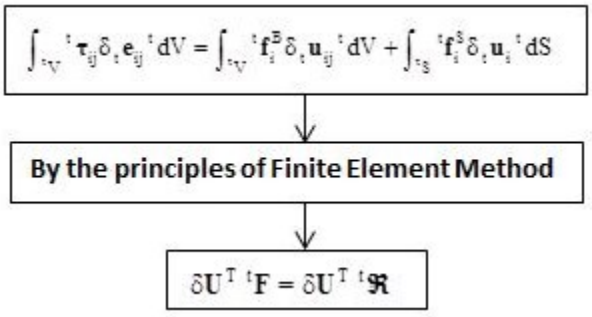


Figure 2.5: Linearization of Principle of Virtual Work

The process of linearization requires the assumption that the solution at time t is known. Hence, ${}^t\boldsymbol{\tau}_i$, ${}^tV$ are known. It is also assumed that the object can undergo large displacements,

large rotations, and large strains. In this incremental approach, the aim is to obtain the solution at time t+Δt. So, applying the PoVW at time t+Δt

$$^{t+\Delta t}\mathbf{F} = {}^{t+\Delta t}\mathfrak{R} \tag{8}$$

to solve for the unknown state at t+Δt, we assume

$$^{t+\Delta t}\mathbf{F} = {}^{t}\mathbf{F} + {}^{t}\mathbf{K}\Delta\mathbf{U}\,. \tag{9}$$

Hence, we solve

$$^{t}\mathbf{K}\Delta\mathbf{U} = {}^{t+\Delta t}\mathbf{R} - {}^{t}\mathbf{F} \tag{10}$$

and obtain,

$$^{t+\Delta t}\mathbf{U} \doteq \qquad J\,. \tag{11}$$

$^{t}\mathbf{K}$: Tangent stiffness matrix of the system corresponding to time t+Δt.

The total external virtual work:

$$^{t+\Delta t}\mathfrak{R} = \int_{^{t+\Delta t}V} {}^{t+\Delta t}\mathbf{f}_i^B \delta\, {}_{t+\Delta t}\mathbf{u}_{ij}\, {}^{t+\Delta t}dV + \int_{^{t+\Delta t}S} {}^{t+\Delta t}\mathbf{f}_i^S \delta\, {}_{t+\Delta t}\mathbf{u}_i\, {}^{t+\Delta t}dS\,. \tag{12}$$

The total internal virtual work:

$$\int_{^{t+\Delta t}V} {}^{t+\Delta t}\boldsymbol{\tau}_{ij} \delta\, {}_{t+\Delta t}\mathbf{e}_{ij}\, {}^{t+\Delta t}dV = {}^{t+\Delta t}\mathfrak{R} \tag{13}$$

$$\delta\, {}_{t+\Delta t}\mathbf{e}_{ij} = \frac{1}{2}\left(\frac{\partial \delta \mathbf{u}_i}{\partial\, {}^{t+\Delta t}\mathbf{x}_j} + \frac{\partial \delta \mathbf{u}_j}{\partial\, {}^{t+\Delta t}\mathbf{x}_i} \right). \tag{14}$$

The equation (12) is a conglomeration of equilibrium, compatibility and stress-strain law at t+Δt. Because, the volume at time $t+\Delta t$ is unknown, it is not possible to integrate over this unknown volume $^{t+\Delta t}V$. Also, it is not possible to directly increment the Cauchy stress. In order to linearize a known reference, configuration is used. The stresses and strains in the current configuration, i.e. Cauchy stress and virtual strain, are transformed into the reference configuration, where they become 2nd Piola Kirchoff stress and Green Lagrange strain tensor. The relationship between 2nd Piola-Kirchoff stress and the Cauchy stress is given by

$$ {}^{t}_{0}\mathbf{S}_{ij} = \frac{{}^{0}\rho}{{}^{t}\rho}\left({}^{0}_{t}\mathrm{X}_{i,m}\, {}^{t}\mathrm{X}_{mn}\, {}^{0}_{t}\mathrm{X}_{i,m} \right) . \tag{15}$$

Since ${}^{t+\Delta t}_{0}\mathbf{S}_{ij}$ and ${}^{t+\Delta t}_{0}\boldsymbol{\varepsilon}_{ij}$ are energetically conjugate, the PoVW is not written as

$$\int_{{}_0V} {}^{t+\Delta t}_{0}\mathbf{S}_{ij}\delta\,{}^{t+\Delta t}_{0}\boldsymbol{\varepsilon}_{ij}\,{}^{0}dV = {}^{t+\Delta t}\mathfrak{R}. \tag{16}$$

Based on the assumption of incremental analysis, the solution at time t is known $({}^{t}_{0}\mathbf{S}_{ij}, {}^{t}_{0}\boldsymbol{\varepsilon}_{ij}, {}^{t}_{0}\mathbf{u}_{ij}...)$. The unknown stresses and strains can be decomposed as follows:

$${}^{t+\Delta t}_{0}\mathbf{S}_{ij} = \underbrace{{}^{t}_{0}\mathbf{S}_{ij}}_{\text{known}} + \underbrace{{}_{0}\mathbf{S}_{ij}}_{\text{unknown stress increment}} \tag{17}$$

$${}^{t+\Delta t}_{0}\boldsymbol{\varepsilon}_{ij} = \underbrace{{}^{t}_{0}\boldsymbol{\varepsilon}_{ij}}_{\text{known}} + \underbrace{{}_{0}\boldsymbol{\varepsilon}_{ij}}_{\text{unknown strain increment}}. \tag{18}$$

Green Lagrange strain at time t and t+Δt is given by

$$\begin{aligned} {}^{t+\Delta t}_{0}\boldsymbol{\varepsilon}_{ij} &= \frac{1}{2}\left({}^{t+\Delta t}_{0}\mathbf{u}_{i,j} + {}^{t+\Delta t}_{0}\mathbf{u}_{j,i} + {}^{t+\Delta t}_{0}\mathbf{u}_{k,i}\,{}^{t+\Delta t}_{0}\mathbf{u}_{k,j} \right), \\ {}^{t+\Delta t}_{0}\boldsymbol{\varepsilon}_{ij} &= \frac{1}{2}\left({}^{t+\Delta t}_{0}\mathbf{u}_{i,j} + {}^{t+\Delta t}_{0}\mathbf{u}_{j,i} + {}^{t+\Delta t}_{0}\mathbf{u}_{k,i}\,{}^{t+\Delta t}_{0}\mathbf{u}_{k,j} \right). \end{aligned} \tag{19}$$

Hence, we find that the strain increment is:

$${}_{0}\boldsymbol{\varepsilon}_{ij} = \frac{1}{2}\underbrace{(\quad\quad\quad\quad\quad\quad\quad\quad)}_{\text{linear in } \mathbf{u}_i} + \frac{1}{2}\underbrace{(\quad\quad\quad\quad)}_{\text{nonlinear in } \mathbf{u}_i}. \tag{20}$$

The linear part is denoted by the symbol ${}_{0}\mathbf{e}_{ij}$, and the nonlinear term is denoted by ${}_{0}\boldsymbol{\eta}_{ij}$. So,

$${}_{0}\boldsymbol{\varepsilon}_{ij} = {}_{0}\mathbf{e}_{ij} + {}_{0}\boldsymbol{\eta}_{ij} \quad \text{and} \quad \delta\,{}_{0}\boldsymbol{\varepsilon}_{ij} = \delta\,{}_{0}\mathbf{e}_{ij} + \delta\,{}_{0}\boldsymbol{\eta}_{ij} \tag{21}$$

We note that the variation in the Green Lagrange strain tensor from 0 to t+Δt, $\delta\,{}^{t+\Delta t}_{0}\boldsymbol{\varepsilon}_{ij}$ is actually $\delta\,{}_{0}\boldsymbol{\varepsilon}_{ij}$. This makes sense physically, because the variation is taken on the displacements at time with fixed ${}^{t}\mathbf{u}_{i}$. (The variation is only imposed on the configuration at time t+Δt, show in Figure 5.1.) Therefore, the PoVW is rewritten as:

$$\int_{{}_0V} {}^{t+\Delta t}_{0}\mathbf{S}_{ij}\delta\,{}_{0}\boldsymbol{\varepsilon}_{ij}\,{}^{0}dV = {}^{t+\Delta t}\mathfrak{R}. \tag{22}$$

On substituting the equations (21), (22) in (16)

$$\underbrace{\int {}_0^t\mathbf{S}_{ij}\,\delta\,{}_0\boldsymbol{\varepsilon}_{ij}\,{}^0dV}_{T1}+\underbrace{\int {}_0^t\mathbf{S}_{ij}\,\delta\,{}_0\boldsymbol{\eta}_{ij}\,{}^0dV}_{T2}={}^{t+\Delta t}\mathfrak{R}-\underbrace{\int {}_0^t\mathbf{S}_{ij}\,\delta\,{}_0\mathbf{e}_{ij}\,{}^0dV}_{T3}. \tag{23}$$

Given the variation, the righthand side is known. The left-hand side contains the unknown displacement increments. It is important to note that no approximations have been made till this point. This equation is, in general, a complicated nonlinear function in the unknown displacement increment. An approximate equation is obtained by neglecting the higher order terms in $\mathbf{u}_i$. This process of neglecting the higher order terms to obtain computationally feasible equation is called linearization. Proceeding to linearization, every individual term in the equation a is treated separately:

T1: Contains terms which are linear in $\mathbf{u}_i$ and nonlinear in $\mathbf{u}_i$.

- ${}_0\mathbf{S}_{ij}$ is a nonlinear function of $\delta\,{}_0\boldsymbol{\varepsilon}_{ij}$
- $\delta\,{}_0\boldsymbol{\varepsilon}_{ij}=\delta\,{}_0\mathbf{e}_{ij}+\delta\,{}_0\boldsymbol{\eta}_{ij}$ is a linear function of $\mathbf{u}_i$

All the higher order terms will be neglected for the same reasons mentioned above.

T2: is linear in $\mathbf{u}_i$.

- ${}_0^t\mathbf{S}_{ij}$ does not contain $\mathbf{u}_i$
- $\delta\,{}_0\boldsymbol{\eta}_{ij}=\frac{1}{2}\left({}_0\delta\mathbf{u}_{k,i}\,{}_0\mathbf{u}_{k,j}+{}_0\mathbf{u}_{k,i}\,{}_0\delta\mathbf{u}_{k,j}\right)$ is linear in $\mathbf{u}_i$

T3: is linear in $\mathbf{u}_i$.

- ${}_0^t\mathbf{S}_{ij}$ does not contain $\mathbf{u}_i$
- $\delta\,{}_0\boldsymbol{\varepsilon}_{ij}=\delta\,{}_0\mathbf{e}_{ij}+\delta\,{}_0\boldsymbol{\eta}_{ij}$, $\delta\,{}_0\mathbf{e}_{ij}$ does not contain $\mathbf{u}_i$ and $\delta\,{}_0\boldsymbol{\eta}_{ij}=\frac{1}{2}\left({}_0\delta\mathbf{u}_{k,i}\,{}_0\mathbf{u}_{k,j}+{}_0\mathbf{u}_{k,i}\,{}_0\delta\mathbf{u}_{k,j}\right)$ is linear in $\mathbf{u}_i$

Hence, the objective is to linearize T2. Expanding the stress increment ${}_0\mathbf{S}_{ij}$ about ${}_0\boldsymbol{\varepsilon}_{rs}$, using Taylor series expansion,

$$ {}_0\mathbf{S}_{ij} = \underbrace{\left.\frac{\partial\, {}_0^t\mathbf{S}_{ij}}{\partial\, {}_0^t\boldsymbol{\varepsilon}}\right|}_{\text{known}} \qquad {}_0\boldsymbol{\varepsilon}_{rs} \qquad + \text{ higher-order terms} \tag{24} $$

$$ \doteq \underbrace{\qquad\qquad}_{\text{known}} \qquad \underbrace{\qquad\qquad}_{\text{in } \mathbf{u}_i \quad \text{quadratic in } \mathbf{u}_i} \doteq \underbrace{\qquad\qquad}_{\text{linearized term}} . \tag{25} $$

Now, substituting this approximated stress increment into T2,

$$ {}_0\mathbf{S}_{ij}\delta_0\boldsymbol{\varepsilon}_{ij} \doteq {}_0\mathbf{e}_{rs}\left(\delta_0\mathbf{e}_{ij}+\delta_0\boldsymbol{\eta}_{ij}\right) \tag{26} $$

$$ \doteq {}_0\mathbf{e}_{rs}\ \delta_0\mathbf{e}_{ij} + {}_0\mathbf{C}_{ijrs}\,{}_0\mathbf{e}_{rs}\ \delta_0\boldsymbol{\eta}_{ij} \qquad \underbrace{\qquad\qquad}_{\text{does not}} \qquad \underbrace{\qquad\qquad}_{\text{quadratic in } \mathbf{u}_i} \tag{27} $$

$$ \doteq \underbrace{\qquad\qquad}_{\text{linearized result}} . \tag{28} $$

The final linearized equation

$$ \underbrace{\int \mathbf{C}\ \mathbf{e}\ \delta\mathbf{e}\ dV + \int {}^t\mathbf{S}\ \delta\boldsymbol{\eta}\ {}^0dV}_{\delta\mathbf{U}^{\cdot}\,{}_0^t\mathbf{K}\Delta\mathbf{U}} = \underbrace{{}^{t+\Delta t}\mathfrak{R} - \int {}^t\mathbf{S}\ \delta\mathbf{e}\ {}^0dV}_{\delta\mathbf{U}^{\cdot}\left({}^{t+\Delta t}\mathfrak{R}-{}_0^t\mathbf{F}\right)} . \tag{29} $$

The representation written below the main equation is obtained after discretizing, using finite element methods. It can be observed that an iteration counter *k* has been introduced to the discretized equation. After each iteration, the error reduces, and the solution approaches the actual solution. Setting a strict convergence criterion neutralizes the many approximations made in the previous steps. The algorithm for carrying out this process is shown in the Figure 2.6.

$$ \int_{{}_0V} {}_0\mathbf{C}_{ijrs}\Delta_0\mathbf{e}_{rs}^{(k)}\delta_0\mathbf{e}_{ij}\ {}_0dV + \int_{{}_0V} {}_0^t\mathbf{S}_{ij}\delta_0\Delta\boldsymbol{\eta}_{ij}^{(k)}\ {}^0dV = {}^{t+\Delta t}\mathfrak{R} - \int_{{}_0V} {}_0^{t+\Delta t}\mathbf{S}_{ij}^{(k-1)}\delta\ {}_0^{t+\Delta t}\boldsymbol{\varepsilon}_{ij}^{(k-1)}\ {}^0dV \tag{30} $$

Again, when discretized:

$$ {}_0^t\mathbf{K}\Delta\mathbf{U}^{(k)} = {}^{t+\Delta t}\mathfrak{R} - {}_0^{t+\Delta t}\mathbf{F}^{(k-1)} \qquad \text{for } k=1, 2, 3.. \tag{31} $$

$$ {}^{t+\Delta t}\mathbf{U}^{(k)} = {}^t\mathbf{U} + \sum_{j=1}^{k}\Delta\mathbf{U}^{(j-1)} \tag{32} $$

$$
\begin{bmatrix} {}^{t}\mathbf{K} \\ \text{nxn} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{U}^{(k)} \\ \text{nx1} \end{bmatrix} = \begin{bmatrix} {}^{t+\Delta t}\mathbf{R} \\ \text{nx1} \end{bmatrix} - \begin{bmatrix} {}^{t+\Delta t}_{0}\mathbf{F}^{(k-1)} \\ \text{nx1} \end{bmatrix} . \tag{33}
$$

The above equation is valid for a single element or an assemblage of elements, where *n* is the total number of degrees of freedom.

Computationally, it is not possible to perform integration. Instead, an approximate value is obtained by using numerical integration algorithms, which are employed to evaluate the integral in thePoVW. In isoparametric finite element analysis of solids, the finite element internal displacements depend on the nodal point displacements. In other words, the displacements at the nodes are only evaluated; the displacement of all other material points is obtained by interpolating the displacements of the neighboring nodes using the function:

$$
{}^{t}\mathbf{u}_{i} = \sum_{k=1}^{N} h_{k}\ {}^{t}\mathbf{u}_{i}^{k} \tag{34}
$$

h_k : Interpolation function of the corresponding node

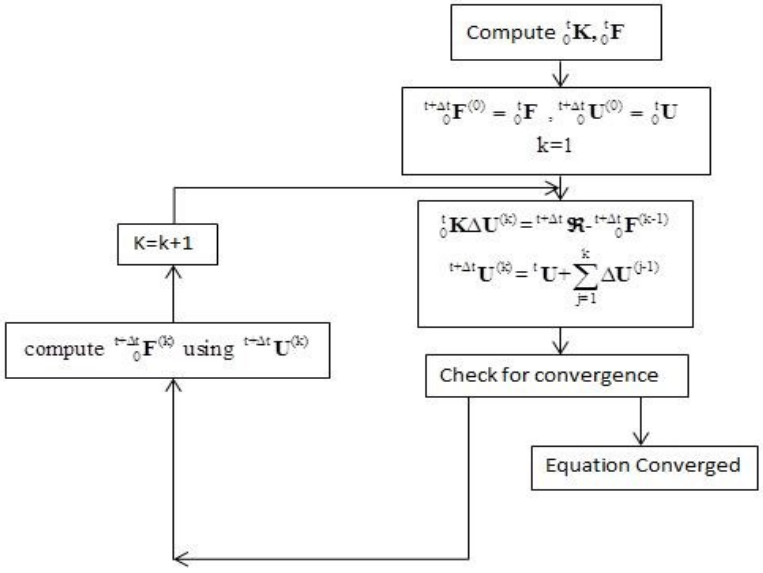


Figure 2.6: Algorithm for general FEM

The equation (31) is the final linearized and discretized form of PoVW, which is used in finite element procedures. This general algorithm is applied to solids undergoing large

deformation and large rotation. In the following sections, this procedure will be tweaked to model the kinematic behavior of Rigid Origami.

In the formulation for RO, deformation is assumed to be small. Therefore, a lot of non-linearity is overcome. One might think that this assumption could produce less accurate results, but the way different elements are coupled in the formulation compensates for the loss in accuracy.

Note:

1. In the above procedure it was assumed the external loads were deformation independent. If inertia was taken into consideration, the equation (12) must be changed as

$$\mathbf{M}\ddot{\ } \quad \zeta\Delta\mathbf{U}^{(k)} = {}^{t+\Delta t}\mathfrak{R} - {}^{t+\Delta t}_{0}\mathbf{F}^{(k-1)} . \tag{35}$$

Here, $\mathbf{M}$ denotes the mass matrix and $\ddot{\ }$ is the acceleration. The analysis is a static/equilibrium if $\ddot{\ }$ =0, and dynamic otherwise.

2. The equation (30) does not entirely represent Green Lagrange strain when the body also experiences large rotation. Some of the non-linear terms corresponding to rotation have been neglected. It can be seen from the equation (29) that, the value of ***K*** depends on these non-linear terms. Hence, the matrix obtained is only an approximate value, but the error is minimized by iterating. On the other hand, the evaluation of *F* only depends on the liner part of the Green Lagrange strain tensor. Hence, its value is always accurate.

3. The current formulation is called a Lagrangian approach because the configuration at t = 0 is used. Similar results will be obtained by using the configuration at time t, called the Eulerian approach. Each has its own advantages and disadvantages.

CHAPTER 3 : DEVELOPMENT OF NON-LOCAL FINITE ELEMENT MODEL

The non-local finite element model (NFEM) was developed on the Atomic Finite Element Method for molecular mechanics simulations. The nodes or vertices of the RO exhibited behavior similar to an atom. This behavior will be discussed in detail in this section.

3.1 NON-LOCAL FINITE ELEMENT METHOD

Before talking about the formulation, it is imperative to understand, "What is non-local deformation?" Consider a pair of atoms. The energy in the atomic bond between them not only depends on the position of the two atoms but also on the relative positions of all the atoms with which these two atoms are associated with. In other words, the position of an atom is dependent on the relative position of all of its first neighbors, and sometimes this dependence might extend farther. This "non-local" dependence of energy is inconsistent with macroscopic, local constitutive models in the conventional FEM. This behavior was observed between the neighboring nodes of an RO. Hence, a macroscopic FEM was developed by drawing inspiration from the Atomic Finite Element Method (AFEM).

In FEM, a continuum is discretized into elements. It is enough to write a program to capture the property of a single element, which can then be used to analyze an assemblage of elements. In this model, the emphasis is to focus on the geometrical properties; material properties are left for future work. Finite element software called ABAQUS, that lets the user define their elements using a subroutine called User Elements (UEL), is employed for this purpose. A detailed discussion about various input and output parameters of a UEL are discussed in the later sections.

3.2 FORMULATION OF NFEM

This section discusses the continuum mechanics aspect of the NFEM. Unlike conventional element types in a finite element analysis, each RO has a unique element type. This number of nodes in each element type is decided based on various factors, such as number of immediate neighboring nodes to the central node or number of nodes connected to the central node via the dihedral angles.

Since material properties of the element are not considered, the formulation is slightly different from the one shown in section. The rigid faces forming the origami are assumed to be rigid, i.e., they suffer very little deformation that, for practical purposes, can be rounded to zero. Instead of starting off at PoVW, an energy based approach is used to derive the governing equation. But the basic structure of the governing equation and the incremental approach is still preserved, i.e.

$$^{t}\mathbf{K}\Delta\mathbf{U} = {}^{t+\Delta t}\mathbf{R} - {}^{t}\mathbf{F} \tag{36}$$

$$^{t+\Delta t}\mathbf{U} \doteq \qquad \text{J}. \tag{37}$$

There are two types of stiffness in any RO. The first one, bond stiffness, prevent any change in the crease length between any two vertices. The second one, angular stiffness, prevents any angular changes between the planar faces. It is assumed that these creases are very rigid and their deformation is very little compared to the angular deformation.

Whenever the system strays from its equilibrium configuration, its internal energy increases; this increase is directly contributed to by angular stiffness. The total energy in an element is written as

$$\mathrm{U}_{\mathrm{tot}} = \sum_{i=1}^{\mathrm{n_{TD}}} \sum_{j=1}^{\mathrm{n_{TE}}} \frac{1}{2} \mathrm{k}_i \left({}_i\alpha_j - {}_i\alpha_{\mathrm{eq}} \right)^2 \tag{38}$$

$\mathrm{U}_{\mathrm{tot}}$: Total Elastic Strain Energy in the element

k_i : Stiffness of dihederal angle along direction i

${}_i\alpha_j$: j^{th} dihederal angle along direction i

${}_i\alpha_{\mathrm{eq}}$: Equilibrium dihederal angle along direction i

$\mathrm{n_{TD}}$: Number of types of dihederal angles in the element

$\mathrm{n_{TE}}$: Total number of dihederal angles in the element

The total energy of the system is given by

$$\mathrm{E}_{\mathrm{tot}}\left(x\right)=\mathrm{U}_{\mathrm{tot}}\left(x\right)-\sum_{i=1}^{\mathrm{n_E}}\mathrm{F}_i\ .\ x_i \tag{39}$$

where $\boldsymbol{x}\in\left[x_1,x_2,...,x_{\mathrm{n_E}}\right]$, and F_i is the external force (if any) applied on the element. As discussed in the section …, for a given set of boundary conditions, an RO system has a stable configuration (equilibrium configuration) where its energy is at its minimum.

$$\frac{\partial \mathrm{E}_{\mathrm{tot}}}{\partial x}=0 \tag{40}$$

$\mathrm{E}_{\mathrm{tot}}$: Total energy of the system

x : Position variable

Let us say that the initial (equilibrium) configuration is ${}^0\boldsymbol{x}\in\left[{}^0x_1,{}^0x_2,...,{}^0x_{\mathrm{n_E}}\right]$, being the initial guess. Expanding $\mathrm{E}_{\mathrm{tot}}$ around ${}^0\boldsymbol{x}$, using the Taylor series expansion yields

$$\mathrm{E}_{\mathrm{tot}}\left(x\right)\doteq \qquad \boldsymbol{x}+\left.\frac{\partial \mathrm{E}_{\mathrm{tot}}}{\partial x}\right|_{\boldsymbol{x}={}^0\boldsymbol{x}}\cdot\left(\boldsymbol{x}-{}^0\boldsymbol{x}\right)+\frac{1}{2}\left(\boldsymbol{x}-{}^0\boldsymbol{x}\right)^T\cdot\left.\frac{\partial^2 \mathrm{E}_{\mathrm{tot}}}{\partial x^2}\right|_{\boldsymbol{x}={}^0\boldsymbol{x}}\cdot\left(\boldsymbol{x}-{}^0\boldsymbol{x}\right)+... \tag{41}$$

On substituting equation (40) into equation (39) and ignoring the higher order terms yield

$$\frac{\partial^2 \mathrm{E}_{\mathrm{tot}}}{\partial x^2}\cdot\left(\boldsymbol{x}-{}^0\boldsymbol{x}\right)=\frac{\partial \mathrm{E}_{\mathrm{tot}}}{\partial x} \tag{42}$$

and

$$\begin{aligned}\frac{\partial^2 \mathrm{E}_{\mathrm{tot}}}{\partial x^2}&=\mathbf{K}\\ &\ .\\ \frac{\partial \mathrm{E}_{\mathrm{tot}}}{\partial x}&=\mathbf{P}\end{aligned} \tag{43}$$

Hence, it can be seen that the governing equation is same as before. Only the evaluation of the stiffness matrix, $\mathbf{K}$, and the out of balance force, $\mathbf{P}$, has changed. From equation (38), it can followed that

$$\mathbf{K} = \frac{\partial^2 \mathrm{U}_{\mathrm{tot}}}{\partial x^2} \tag{44}$$

$$\mathbf{P} = \mathbf{F} - \frac{\partial \mathrm{U}_{\mathrm{tot}}}{\partial x}. \tag{45}$$

$\mathbf{F} \in \left[F_1, F_2, ..., F_{\mathrm{n_E}} \right]$ is the external force applied, and $\frac{\partial \mathrm{U}_{\mathrm{tot}}}{\partial x}$ corresponds to the internal force in the element. Except for linear cases, the equation is iteratively solved for $\mathbf{P} = 0$ till a convergence criterion is satisfied. The Newton Raphson iterative scheme is used to achieve convergence, which will be discussed in detail in the forth coming section.

3.3 NFEM ELEMENT

It is important to note that NFEM elements are pattern specific, i.e. they are different for different origami. They depend on the primary structure and also on the interactions (e.g., nearest neighbor and second nearest neighbor interactions). Consider a general NFEM element with *n* nodes in space as shown in Figure 3.1. Let the central atom have $\mathrm{n_p}$ nearest neighbors and $\mathrm{n_s}$ second nearest neighbors; which means, $\mathrm{n} = \mathrm{n_p} + \mathrm{n_s}$. The stiffness matrix and out of balance force vector is given as

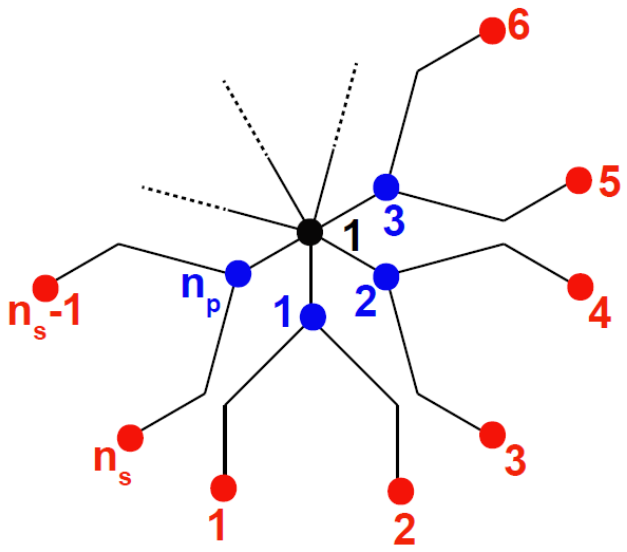


Figure 3.1: General Non-local Finite Element

$$\mathbf{K}^e = \begin{bmatrix} \left(\dfrac{\partial^2 \mathrm{U}_{\mathrm{tot}}}{\partial x_1 \partial x_1}\right)_{3\mathrm{x}3} & \left(\dfrac{1}{2}\dfrac{\partial^2 \mathrm{U}_{\mathrm{tot}}}{\partial x_1 \partial x_i}\right)_{3\mathrm{x}(\mathrm{n}-1)} \\ \left(\dfrac{1}{2}\dfrac{\partial^2 \mathrm{U}_{\mathrm{tot}}}{\partial x_i \partial x_1}\right)_{(\mathrm{n}-1)\mathrm{x}3} & (0)_{(\mathrm{n}-1)\mathrm{x}(\mathrm{n}-1)} \end{bmatrix} \quad \mathbf{P}^e = \begin{bmatrix} \left(\mathbf{F} - \dfrac{\partial \mathrm{U}_{\mathrm{tot}}}{\partial x_1}\right)_{3\mathrm{x}1} \\ (0)_{(\mathrm{n}-1)\mathrm{x}1} \end{bmatrix} \tag{46}$$

$\mathbf{K}^e$: Elemental stiffness matrix $\quad i$ ranges from 2 to n

$\mathbf{P}^e$: Elemental out of balance force vector

$\mathbf{F}$: External force vector

The above stiffness matrix is very different from its conventional counterpart. The NFEM stiffness matrix focuses on the central node; as a result, the matrix is sparse. When all the stiffness matrices are assembled into the global stiffness matrix, the non-local property with respect to each central node is captured.

The overlap in the NFEM element does not double count the element contribution to the global stiffness matrix $\mathbf{K}$ and non-equilibrium force vector $\mathbf{P}$. In order to prevent the double count, a factor of ½ is introduced in the non-diagonal elements. The matrices in the equation (44) are accurate, because they do not involve any approximations of conventional FEM, such as the use of shape functions within the element and the neglect of the non-local effect in a local element. By the standard assembly in FEM, the global stiffness matrix $\mathbf{K}$ and non-equilibrium force vector $\mathbf{P}$ for the system are readily obtained from their counterparts at the element level. Contrary to conventional FEM, the i^{th} row of the global stiffness matrix $\mathbf{K}$ and the i^{th} component of the non-equilibrium force vector $\mathbf{P}$ are completely determined by this single AFEM element for atom *i.* Although the AFEM elements overlap in space, there is no double count of contributions to $\mathbf{K}$ and $\mathbf{P}$ from AFEM elements.

It has to be pointed out that, the number of non-zeros entries is less than or equal to $n^2 - (n-1)^2$. This can be simplified to $2n-1$, which is of the order *n*. For order-*n* sparse $\mathbf{K}$, the computational effort to solve $\mathbf{Ku} = \mathbf{P}$ is also on the order of *n*.

3.4 Difference between conventional FEM and NFEM

In conventional finite elements, as discussed in the section 2.4 the stiffness matrix and the out of balance force vector are evaluated after every iteration, from PoVW. The Continuum is discretized into elements that don't overlap. The total system energy is dispersed into the elements. The elemental matrices are based off of the nodal displacement.

In the case of NFEM, $\mathbf{K}$ and $\mathbf{P}$ can be obtained straightforwardly via the continuum conventional FEM, since the interaction between two adjacent nodes can be equivalently represented by a nonlinear string element. For multi-body interaction, however, the energy is influenced by nodes which are not part of the pair. A conventional FEM cannot capture this non-local dependence of energy, since it requires information beyond the element. Unlike conventional continuum FEM, the energy is not divided into elements. $\mathbf{K}$ and $\mathbf{P}$ can be directly obtained by differentiating total energy of the system E_{tot} needed in the governing equation (41), directly from this single AFEM element.

3.5 Stability of NFEM

Instability may occur due to geometric nonlinearities. The stability and convergence are ensured if the energy in the system decreases in every step, i.e., $\mathbf{u} \cdot \mathbf{P} > \mathbf{0}$. It represents the steepest descent direction of E_{tot}. A sufficient condition for $\mathbf{K} \cdot \mathbf{u} > \mathbf{0}$ is that the stiffness matrix $\mathbf{K}$ is positive definite. For problems involving neither material softening nor nonlinear bifurcation, $\mathbf{K}$ is usually positive definite and NFEM is stable. For non-positive definite $\mathbf{K}$, $\mathbf{K}$ is replaced by $\mathbf{K}^* = \mathbf{K} + \alpha\mathbf{I}$, where **I** is the identity matrix and α is a positive number to ensure the positive definiteness of $\mathbf{K}^*$, which guarantees $\mathbf{u} \cdot \mathbf{P} > \mathbf{0}$. It is important to note that the state of minimal energy is independent of α. This is because the energy minimum is characterized by $\mathbf{P} = \mathbf{0}$ and independent of $\mathbf{K}$ or $\mathbf{K}^*$. In fact, at (and near) the state of minimum energy, such modification of $\mathbf{K}$ is unnecessary. because the stiffness matrix K becomes positive definite.

CHAPTER 4 : NON LOCAL FINITE ELEMENTS

One advantage of NFEM is that it has same theoretical framework as conventional FEM. NFEM element has been implemented for a couple of RO patterns (Miura-Ori and Ron Resch) in the ABAQUS finite element program via its USER-ELEMENT subroutine (UEL), and several of their geometrical properties have been studied. This section discusses the details of the development of UEL program.

The first step in developing a UEL for an element is the identification of the nearest neighbors and second nearest neighbors. The next step involves the identification of different types of dihedral angles that connect the central node and the second nearest neighbors. Once these are known, the relationship between various angles is determined. These steps are common for all elements; only the details vary from element to element. Below, the UEL procedure for development procedure for a couple of elements, namely Miura-Ori and Ron Resch, are discussed.

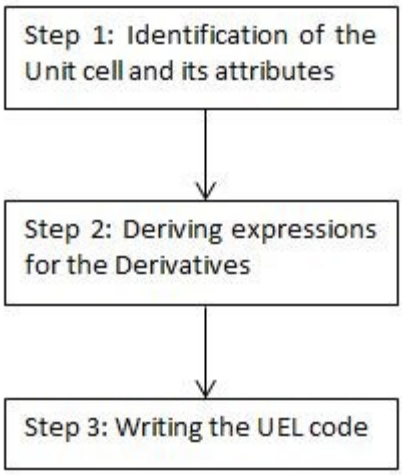


Figure 4.1: Algorithm for Non-local Finite Element Method

4.1 MIURA-ORI

The Miura-Ori is a method of folding a flat surface, such as a sheet of paper, into a smaller volume. It is named for Japanese astrophysicist Koryo Miura, who invented it.

The crease patterns of the Miura fold form a tessellation of the surface by parallelograms. In one direction, the creases of the pattern lie along straight lines, with each parallelogram

forming the mirror reflection of its neighbor across each crease. In the other direction, the creases zigzag, and each parallelogram is the translation of its neighbor across the crease. Each of the zigzag paths of creases consists solely of mountain folds or of valley folds, with mountains alternating with valleys from one zigzag path to the next. Each of the straight paths of creases alternates between mountain and valley folds.

A folded Miura fold can be packed into a very compact shape, its thickness restricted only by the thickness of the folded material. The fold can also be unpacked in just one motion by pulling on opposite ends of the folded material and likewise folded again by pushing the two ends back together. In the application to solar arrays, this property reduces the number of motors required to unfold this shape, reducing the overall weight and complexity of the mechanism.

4.1.1 Step 1: Identification of the MO Unit Cell and its Attributes

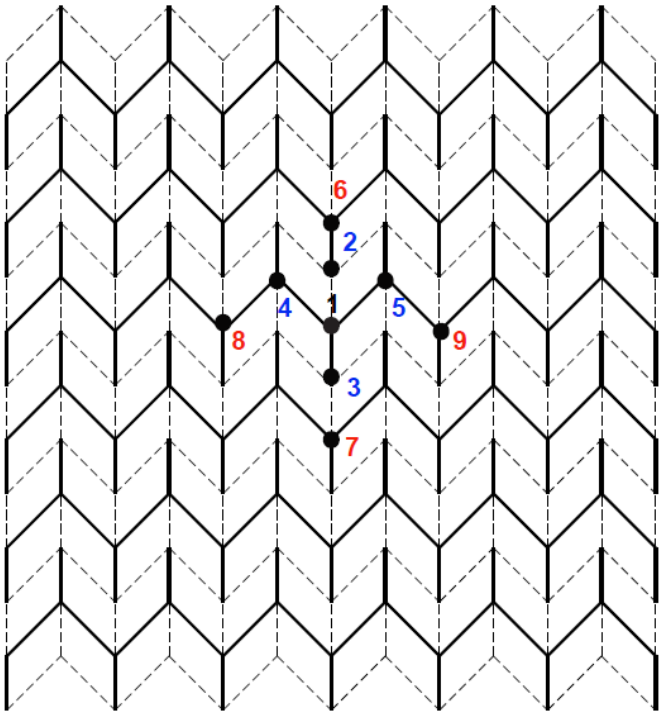


Figure 4.2: 2D crease pattern for Miura-Ori

Here the primary and secondary neighbors are shown with different colors. Some key features are listed below in the Table 4.1.

Table 4.1: Type I Element of Miura-Ori

Dihedral Angles		*Rigid bonds*	
Angle #1	(1,5,9), (1,4,8), (4,1,5)	$L = R$	(1,3), (3,7), (1,4), (4,8), (1,5), (5,9), (1,2), (2,6)
Angle #2	(1,3,7),(1,2,6),(2,1,3)		

4.1.2 STEP 2: EXPRESSIONS FOR THE DIHEDRAL DERIVATIVES OF MO

The entire problem boils down to solving the equation $\mathbf{Ku} = \mathbf{P}$, which requires the determination of the first and second derivatives of the total energy U_{tot} with respect to coordinates. But the total energy as seen in equation (37) is a function of the dihedral angle. So, it is necessary to find the formula for the dihedral angles, either in terms of some other fundamental angle or directly from the coordinates of the nodes.

The following section discusses the derivation of various required relationships and discusses how to determine the first and second derivate required for developing the UEL.

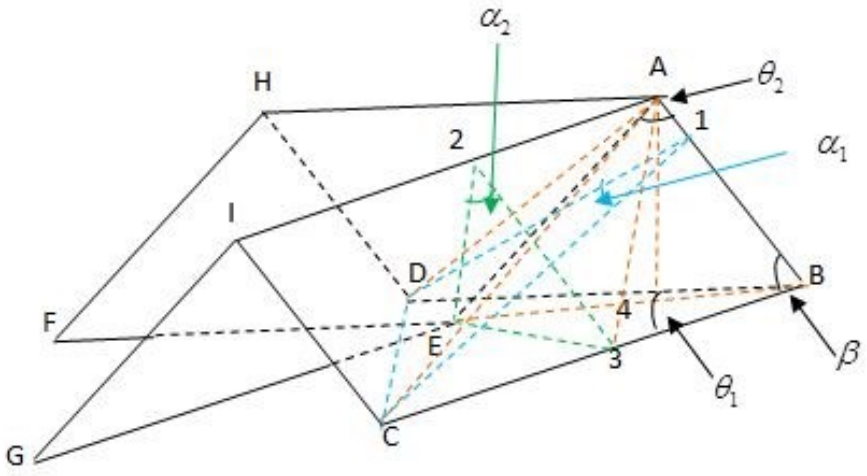


Figure 4.3: Miura-Ori Unit cell

Figure 4.3 shows the unit cell of Miura-Ori. This pattern is periodic in nature; in other words, the entire structure at the un-deformed state can be obtained by duplicating the unit cells in X and Y directions.

Assumptions:

1. Points B, C, D, E, F, and G are on the same plane.

2. Points A, I, and H are on the same plane.

3. Lines 1D $\perp$ AB and 1C $\perp$ AB.

4. Lines 2E $\perp$ AI, 23 $\perp$ AI.

5. Lines A4 $\perp$ BE and A4 $\perp$ plane ABC.

Table 4.2: Notations used in MO formulation

θ_1	Angle $\angle(\mathrm{D,B,C})$
θ_2	Angle $\angle(\mathrm{E,1,B})$
α_1	Dihedral angle, $\angle(\mathrm{D,1,C})$
α_2	Dihedral angle, $\angle(\mathrm{E,2,3})$
β	Acute face (parallelogram) angle
a	Length of crease AB
b	Length of crease BC

The unit cell shown in Figure 4.3 is very simple and has a single controlling parameter ($\angle(\mathrm{F,E,G})$). The equations for the dihedral angles can be directly obtained in terms of β and θ. Here, θ can take up the value of θ_1 or θ_2, depending on the instance in the simulation.

DIHEDRAL ANGLES α_1, and α_2 :

The expressions for the three dihedral angles, α_1, and α_2 in terms of θ and β, are presented below,

α_1 : $\Rightarrow$ First dihedral angle is between planes (A,B,C,I) and (A,B,D,H)

Applying the law of cosines in the triangle $(\mathrm{D,1,C})$,

$$\alpha_1 = \arccos\left(\frac{|1D|^2 + |1C|^2 - |DC|^2}{2|1D||1C|}\right), \tag{47}$$

$$= \arccos\left(\frac{b^2 \sin^2\beta + b^2 \sin^2\beta - 4b^2 \sin^2 \frac{\theta_1}{2}}{2b^2 \sin^2\beta} \right), \tag{48}$$

$$\alpha_1 = \arccos\left(1 - 2\frac{\sin^2 \frac{\theta_1}{2}}{\sin^2\beta} \right). \tag{49}$$

α_2 : $\Rightarrow$ Second dihedral angle is between planes (A,B,C,I) and (A,E,G,I)

The volume of the tetrahedral ABCD can be written as:

$$V_{ABCD} = \frac{1}{2}|AB|\, A_{D1C} = \frac{1}{2}a \cdot \frac{1}{2} 2b^2 \sin\frac{\theta_1}{2} \sqrt{\sin^2\beta - \sin^2\frac{\theta_1}{2}}\,. \tag{50}$$

It can also be written as:

$$V_{ABCD} = \frac{1}{2}|A4|\, A_{BCD} = \frac{1}{2}|A4| \cdot \frac{1}{2} 2b^2 \sin\frac{\theta_1}{2}\cos\frac{\theta_1}{2} \tag{51}$$

Equating equations (47) and (48), we get

$$|A4| = \frac{\sqrt{\sin^2\beta - \sin^2\frac{\theta_1}{2}}}{\cos\frac{\theta_1}{2}} a\,. \tag{52}$$

Applying the law of cosines in triangle $(\mathrm{E,A,B})$,

$$\angle \mathrm{EAB} = \theta_2 = \arccos\left(\frac{|AE|^2 + |AB|^2 - |EB|^2}{2|AE||AB|} \right). \tag{53}$$

On substituting the values into equation (51)

$$= \arccos\left(\frac{a^2 + a^2 - 4a^2 \frac{\cos^2\beta}{\cos^2\frac{\theta_1}{2}}}{2a^2} \right) \tag{54}$$

$$= \arccos\left(1 - \frac{2\cos^2\beta}{\cos^2\frac{\theta_1}{2}}\right). \tag{55}$$

Also,

$$|EB| = 2a\frac{\cos\beta}{\cos\frac{\theta_1}{2}},\ |3B| = 2a\cos\beta\,. \tag{56}$$

Applying the law of cosines in triangle $(\mathrm{E,B,3})$,

$$|E3| = \sqrt{|EB|^2 + |3B|^2 - 2|EB||3B|\cos\angle EB3}\,. \tag{57}$$

Substituting the equation (54) into equation (55), we get

$$= \sqrt{4a^2\frac{\cos^2\beta}{\cos^2\frac{\theta_1}{2}} + 4a^2\cos^2\beta - 2\cdot 2a\frac{\cos\beta}{\cos\frac{\theta_1}{2}}\cdot 2a\cos\beta\cdot\cos\frac{\theta_1}{2}} \tag{58}$$

$$= 2\cos\beta a\sqrt{\frac{1}{\cos^2\frac{\theta_1}{2}} - 1} \tag{59}$$

$$= 2\cos\beta\tan\frac{\theta_1}{2}a\,. \tag{60}$$

Applying the law of cosines in triangle $(\mathrm{E,2,3})$,

$$\alpha_2 = \angle\mathrm{E23} = \arccos\left(\frac{|2E|^2 + |23|^2 - |E3|^2}{2|2E||23|}\right). \tag{61}$$

Substituting the equation (58) into equation (59),

$$= \arccos\left(\frac{a^2\sin^2\beta + a^2\sin^2\beta - 4a^2\cos^2\beta\tan^2\frac{\theta_1}{2}}{2a^2\sin^2\beta}\right) \tag{62}$$

$$= \arccos\left(1 - 2\cot^2\beta \tan^2\frac{\theta_1}{2}\right). \tag{63}$$

Substituting the equation (53) in to equation (61), the expression for the second dihedral angle is obtained as shown below:

$$\alpha_2 = \arccos\left(\frac{\cos\theta_2 + \cos^2\beta}{\sin^2\beta}\right). \tag{64}$$

Derivatives of the Dihedral angles:

$$\frac{\partial\alpha_1}{\partial\theta_1} = \left(\frac{\sqrt{2}\cos^2\left(\frac{\theta_1}{2}\right)}{\sqrt{\cos(\theta_1) - \cos(2\beta)}}\right) \tag{65}$$

$$\frac{\partial^2\alpha_1}{\partial\theta_1^2} = \left(\frac{\sqrt{2}\cos^2(\beta)}{\sin^2\left(\frac{\theta_1}{2}\right)\left(\cos(\theta_1) - \cos(2\beta)\right)^{\frac{3}{2}}}\right) \tag{66}$$

$$\frac{\partial\alpha_2}{\partial\theta_2} = \left(\frac{\sin(\theta_2)}{\sqrt{\sin^4(\beta) - \left(\cos^2(\beta) + \cos^2(\theta_2)\right)^2}}\right) \tag{67}$$

$$\frac{\partial^2\alpha_2}{\partial\theta_2^2} = \left(\frac{2\cos^2\left(\frac{\theta_2}{2}\right)\cos^2(\beta)}{\left(\cos(2\beta) + \cos(\theta_2)\right)\sqrt{\sin^4(\beta) - \left(\cos^2(\beta) + \cos^2(\theta_2)\right)^2}}\right) \tag{68}$$

4.1.3 STEP 3: WRITING THE UEL CODE FOR MO

The UEL code is attached in the appendix. It has four Subroutines and thirteen functions. A brief description of the three subroutines is given below

1. SUBROUTINE UEL: This is the main subroutine called by ABAQUS while running the program. This will be called for each element that is of a general user-defined element type (i.e., not defined by a linear stiffness or mass matrix read either directly or from a results file data) each

time element calculations are required and must perform all of the calculations for the element, appropriate to the current activity in the analysis.

2. SUBROUTINE CALEMEF: This subroutine will be called by SUBROUTINE UEL to evaluate the 27x27 elemental stiffness matrix and a 27x1 non-equilibrium force vector for each element.

3. DUBROUTINE SPRINGEMEF: This subroutine will be called by SUBROUTINE UEL to evaluate the xx6 spring stiffness matrix and a 6x1 non-equilibrium force vector for each spring element.

4. SUBROUTINE DIFF: This subroutine will be called by SUBROUTINE CALEMEF to evaluate the first and second derivate of the total energy. This is the corner stone of the entire program, as a lot of math needed to be consistent in order for this phase to work. The relationships derived in step 2 will be used in this part to evaluate the partial derivatives of the total energy using the chain rule.

4.2 Ron Resch

This pattern was conceived by Ron Resch, a visionary mathematician and designer who was one of first to explore the architectural potential of 3D tessellated structures in the 1960′s.

4.2.1 Step 1: Identification of the RR Unit cell and its attributes

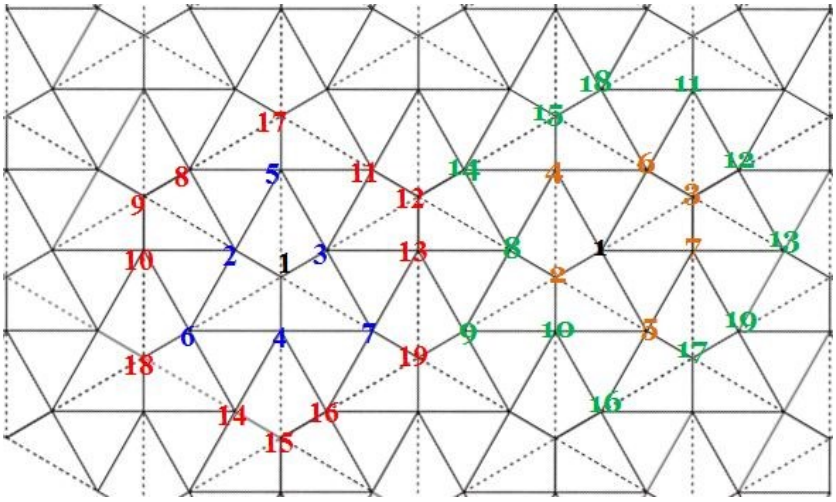


Figure 4.4: 2D Crease pattern for Ron Resch

It can be observed that, unlike in the case of MO, there is not just a single unit cell in this pattern; there are two unit cells which form the types of elements. As a result of this non-periodicity, two nineteen node elements were used in the program. Here, the primary and secondary neighbors are shown with different colors.

Table 4.3: Type I Element of Ron Resch

Dihedral Angles		*Rigid bonds*	
Angle #1	(2,1,3), (2,1,4), (3,1,4)	$L = \frac{R}{\sqrt{3}}$	(1, 2), (1, 3), (1, 4), (5, 17), (6, 18), (7, 19), (8, 9), (9, 10), (11, 12), (12, 13), (14, 15), (15, 16)
Angle #2	(5,1,6), (5,1,7), (6,1,7)		
Angle #3	(2,1,7), (3,1,6), (4,1,5)	$L = R$	(2, 3), (2, 4), (2, 5), (2, 6), (2, 8), (2, 10), (3, 4), (3, 5), (3, 7), (3, 11), (3, 13), (4, 6), (4, 7), (4, 14), (4, 16), (5, 8), (5, 11), (6, 10), (6, 14), (7, 13), (7, 16), (8, 10), (11,13), (14, 16)
Angle #4	(1,2,8), (1,2,10), (1,3,11), (1,3,13), (1,4,14), (1,4,16)		
Angle #5	(1,2,9), (1,3,12), (1,4,15), (1,5,17), (1,6,18), (1,7,19)		
Angle #6	(1,5,8), (1,5,11), (1,6,10), (1,6,14), (1,7,13), (1,7,16)	$L = \frac{2R}{\sqrt{3}}$	(1, 5), (1, 6), (1, 7), (2, 9), (3, 12), (4, 15), (8, 17), (10, 18), (11, 17), (13, 19), (14, 18), (16, 19)

Table 4.4: Type II Element of Ron Resch

Dihedral Angles		*Rigid bonds*	
Angle #1	(1,2,8), (1,2,10)	$L = \frac{R}{\sqrt{3}}$	(1, 2), (2, 8), (2, 10), (3, 6), (3, 7), (3, 12), (4, 15), (5, 17), (15, 18), (17, 19)
Angle #2	(1,3,11), (1,3,13)		
Angle #3	(1,2,9), (1,3,12)	$L = R$	(1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (1, 10), (4, 6), (4, 8), (4, 14), (4, 18), (5, 7), (5, 10), (5, 16), (5, 19), (6, 7), (6, 11), (6, 12), (6, 18), (7, 12), (7, 13), (7, 19), (8, 9), (8, 10), (8, 14), (9, 10), (10, 16), (11, 12), (11, 18), (12, 13), (13, 19)
Angle #4	(2,1,6), (2,1,7), (1,4,15), (1,5,17)		
Angle #5	(2,1,3)		
Angle #6	(3,1,4), (3,1,5), (1,6,15), (1,7,17)		
Angle #7	(4,1,5), (1,6,11), (1,7,13)		
Angle #8	(4,1,7), (5,1,6), (1,4,14), (1,5,16), (1,6,18), (1,7,19)	$L = \frac{2R}{\sqrt{3}}$	(1, 3), (2, 4), (2, 5), (2, 9), (3, 11), (3, 13), (6, 15), (7, 17), (14, 15), (16, 17),
Angle #9	(6,1,7), (1,4,8), (1,5,10)		

4.2.2 STEP 2: EXPRESSIONS FOR THE DIHEDRAL DERIVATIVES OF RR

As said earlier, since total energy as seen in equation (37) is a function of the dihedral angle, it is necessary to find formula for the dihedral angles, either in terms of some other fundamental angle or directly as the coordinates of the nodes.

It is very important to understand that even though there are up to 9 dihedral angles in the element, there are only three types of dihedral angles which relate the central node with the secondary neighbors; the values of these three angles will only be set in the program. So it is imperative to find the relationships of the nine angles in terms of these three angles, so that it is easy to take the derivatives.

The following section discusses the derivation of various required relationships and discusses how to determine the first and second derivate required for developing the UEL.

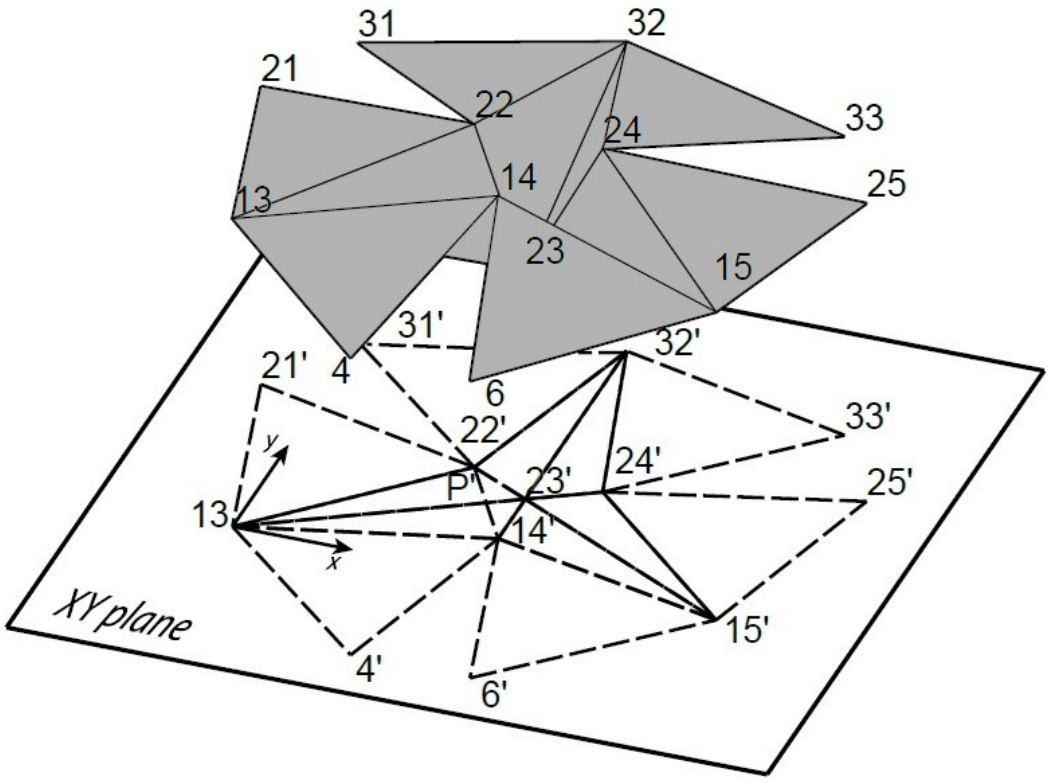


Figure 4.5: Ron Resch unit cell

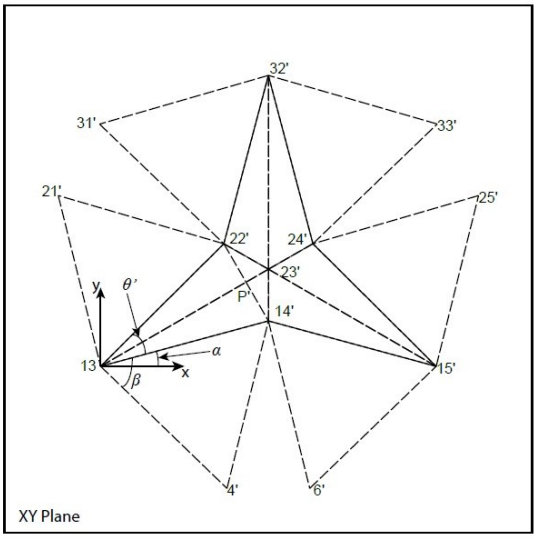


Figure 4.6: Projection of Ron Resch unit cell on XY plane

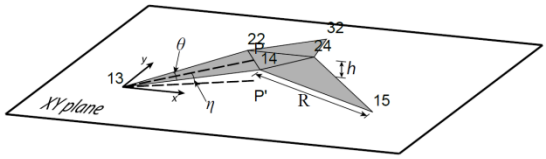


Figure 4.7: Virtual planes in Ron Resch unit cell

Figure 4.5 shows a RR unit cell and the projection of its edges on the XY plane. Note, there is no specific reason for the numbering sequence of the nodes. Figure 4.6 shows the top view of the projection to illustrate the various angles involved in the formulation. Figure 4.7 shows the virtual planes (shown in grey); they do not form a part of the unit cells. These planes are constructed just to determine the relationship between various parameters.

Assumptions:

1: Points 13, 15, and 32 are on the same plane (XY plane).

2. Points 14, 22, and 24 are on the same plane which is parallel to XY plane.

3. Points 4, 25, 31 are on the same plane.

4. Points 6, 21, 33 are on the same plane.

5. Point 23 does not lie on any of the other planes.

Table 4.5: Notations used in Ron Resch formulation

The letter (or symbol) by itself denotes the physical points and angles and, used with ', denotes its projection on XY plane.	
η	Dihedral angle formed by the plane, formed by (14, 22, 24) with XY plane
h	Distance of the plane formed by (14, 22, 24) from XY plane

γ_1	Dihedral angle formed by the plane, formed by (4, 25, 31) with XY plane
γ_2	Dihedral angle formed by the plane, formed by (6, 21, 33) with XY plane
H_1	Distance of the plane formed by (4, 25, 31), from XY plane
H_2	Distance of the plane formed by (6, 21, 33), from XY plane
θ	$\angle(22,13,14)$
R	Rigid bond length of line 4-6 in Fig 7.1
P	Center of line joining (22, 14)
β	$\angle(14',13,4'),\ \angle(14',15,6')$
α	Angle between (13, 14') and X axis, Angle between (13, 22') and Y axis

First, the aim is to determine the coordinates of all the nodes of a unit cell shown in Figure 4.5. From Figure 4.6, it can be observed that

$$2\alpha + \theta' = \frac{\pi}{3} \tag{69}$$

$$|13,P| = R\cos\left(\frac{\theta}{2}\right) \tag{70}$$

$$|22,P| = |22',P'| = |14',P'| = |14,P| = R\sin\left(\frac{\theta}{2}\right) \tag{71}$$

$$|22,14| = |22',14'| = 2|22,P| = 2R\sin\left(\frac{\theta}{2}\right) \tag{72}$$

$$h = |13,P|\sin\eta = R\cos\left(\frac{\theta}{2}\right)\sin\eta \tag{73}$$

$$|13,P'| = |13,P|\cos\eta = R\cos\left(\frac{\theta}{2}\right)\cos\eta \tag{74}$$

$$\tan\left(\frac{\theta'}{2}\right)=\frac{|22',P'|}{|13,P'|}=\frac{R\sin\left(\frac{\theta}{2}\right)}{R\cos\left(\frac{\theta}{2}\right)\cos\eta}=\frac{\tan\left(\frac{\theta}{2}\right)}{\cos\eta} \tag{75}$$

$$\sin\left(\frac{\theta'}{2}\right)=\frac{\tan\left(\frac{\theta}{2}\right)}{\sqrt{\tan^2\left(\frac{\theta}{2}\right)+\cos^2\eta}} \tag{76}$$

$$\cos\left(\frac{\theta'}{2}\right)=\frac{\cos\eta}{\sqrt{\tan^2\left(\frac{\theta}{2}\right)+\cos^2\eta}} \tag{77}$$

$$\sin\theta'=\frac{2\tan\left(\frac{\theta}{2}\right)\cos\eta}{\tan^2\left(\frac{\theta}{2}\right)+\cos^2\eta}=\frac{\sin\theta\cos\eta}{\sin^2\left(\frac{\theta}{2}\right)\sin^2\eta+\cos^2\eta} \tag{78}$$

$$\cos\theta'=\frac{\cos^2\eta-\tan^2\left(\frac{\theta}{2}\right)}{\tan^2\left(\frac{\theta}{2}\right)+\cos^2\eta}=\frac{\cos\theta-\sin^2\eta\cos^2\left(\frac{\theta}{2}\right)}{\sin^2\left(\frac{\theta}{2}\right)\sin^2\eta+\cos^2\eta} \tag{79}$$

Using equation (67),

$$\sin(2\alpha)=\frac{\sqrt{3}}{2}\cos\theta'-\frac{1}{2}\sin\theta'=\frac{\sqrt{3}\left[\cos^2\eta-\tan^2\left(\frac{\theta}{2}\right)\right]-2\tan\left(\frac{\theta}{2}\right)\cos\eta}{2\left[\tan^2\left(\frac{\theta}{2}\right)+\cos^2\eta\right]} \tag{80}$$

$$\cos(2\alpha)=\frac{1}{2}\cos\theta'+\frac{\sqrt{3}}{2}\sin\theta'=\frac{\cos^2\eta-\tan^2\left(\frac{\theta}{2}\right)+\sqrt{3}\left[2\tan\left(\frac{\theta}{2}\right)\cos\eta\right]}{2\left[\tan^2\left(\frac{\theta}{2}\right)+\cos^2\eta\right]} \tag{81}$$

$$\cos\alpha=\frac{\sqrt{3}\cos\eta+\tan\left(\frac{\theta}{2}\right)}{2\sqrt{\tan^2\left(\frac{\theta}{2}\right)+\cos^2\eta}}=\frac{\sqrt{3}\cos\eta\cos\frac{\theta}{2}+\sin\frac{\theta}{2}}{2\sqrt{\sin^2\left(\frac{\theta}{2}\right)\sin^2\eta+\cos^2\eta}} \tag{82}$$

$$\sin\alpha = \frac{\left|\sqrt{3}\tan\left(\frac{\theta}{2}\right)-\cos\eta\right|}{2\sqrt{\tan^2\left(\frac{\theta}{2}\right)+\cos^2\eta}} = \frac{\left|\sqrt{3}\sin\left(\frac{\theta}{2}\right)-\cos\eta\cos\left(\frac{\theta}{2}\right)\right|}{2\sqrt{\sin^2\left(\frac{\theta}{2}\right)\sin^2\eta+\cos^2\eta}} \tag{83}$$

$$\sin\left(\alpha+\frac{\pi}{6}\right) = \sin\alpha\frac{\sqrt{3}}{2}+\cos\alpha\frac{1}{2} = \begin{cases} \dfrac{\sin\left(\frac{\theta}{2}\right)}{\sqrt{\sin^2\left(\frac{\theta}{2}\right)\sin^2\eta+\cos^2\eta}}, & \text{for } \sqrt{3}\tan\left(\frac{\theta}{2}\right) > \cos\eta \\ \dfrac{\sqrt{3}\cos\eta\cos\frac{\theta}{2}-\sin\frac{\theta}{2}}{2\sqrt{\sin^2\left(\frac{\theta}{2}\right)\sin^2\eta+\cos^2\eta}}, & \text{for } \sqrt{3}\tan\left(\frac{\theta}{2}\right) < \cos\eta \end{cases} \tag{84}$$

$$\cos\left(\alpha+\frac{\pi}{6}\right) = \cos\alpha\frac{\sqrt{3}}{2}-\sin\alpha\frac{1}{2} = \begin{cases} \dfrac{\cos\eta\cos\left(\frac{\theta}{2}\right)}{\sqrt{\sin^2\left(\frac{\theta}{2}\right)\sin^2\eta+\cos^2\eta}}, & \text{for } \sqrt{3}\tan\left(\frac{\theta}{2}\right) > \cos\eta \\ \dfrac{\cos\eta\cos\left(\frac{\theta}{2}\right)+\sqrt{3}\sin\left(\frac{\theta}{2}\right)}{2\sqrt{\sin^2\left(\frac{\theta}{2}\right)\sin^2\eta+\cos^2\eta}}, & \text{for } \sqrt{3}\tan\left(\frac{\theta}{2}\right) < \cos\eta \end{cases} \tag{85}$$

$$\begin{aligned} \left|13,14'\right| = \left|13,22'\right| = \frac{\left|14',P'\right|}{\sin\left(\frac{\theta'}{2}\right)} &= \frac{R\sin\left(\frac{\theta}{2}\right)}{\frac{\tan\left(\frac{\theta}{2}\right)}{\sqrt{\tan^2\left(\frac{\theta}{2}\right)+\cos^2\eta}}} = R\cos\left(\frac{\theta}{2}\right)\sqrt{\tan^2\left(\frac{\theta}{2}\right)+\cos^2\eta} \\ &= R\sqrt{\sin^2\left(\frac{\theta}{2}\right)\sin^2\eta+\cos^2\eta} \end{aligned} \tag{86}$$

Now based on the above relationships, we move on to calculate the coordinates of various nodes in terms of various angles.

Coordinates of point 13: $\Rightarrow$

$$x_{13}=0, y_{13}=0, z_{13}=0 \tag{87}$$

Coordinates of point 15: $\Rightarrow$

$$\begin{aligned} x_{15} &= 2\left|13,14'\right|\cos\alpha = R\left(\sqrt{3}\cos\eta\cos\frac{\theta}{2}+\sin\frac{\theta}{2}\right), \\ y_{15} &= 0, \\ z_{15} &= 0 \end{aligned} \tag{88}$$

Coordinates of point 32: $\Rightarrow$

$$\begin{aligned} x_{32} &= \frac{x_{15}}{2} = \frac{R}{2}\left(\sqrt{3}\cos\eta\cos\frac{\theta}{2}+\sin\frac{\theta}{2}\right), \\ y_{32} &= \frac{\sqrt{3}}{2}x_{15} = \frac{\sqrt{3}R}{2}\left(\sqrt{3}\cos\eta\cos\frac{\theta}{2}+\sin\frac{\theta}{2}\right), \\ z_{32} &= 0 \end{aligned} \tag{89}$$

Coordinates of point 14: $\Rightarrow$

$$\begin{aligned} x_{14} &= x_{32} = \frac{R}{2}\left(\sqrt{3}\cos\eta\cos\frac{\theta}{2}+\sin\frac{\theta}{2}\right), \\ y_{14} &= \left|13,14'\right|\sin\alpha = \frac{R}{2}\left|\sqrt{3}\sin\frac{\theta}{2}-\cos\eta\cos\frac{\theta}{2}\right|, \\ z_{14} &= h = R\cos\left(\frac{\theta}{2}\right)\sin\eta \end{aligned} \tag{90}$$

Coordinates of point 22: $\Rightarrow$

$$\begin{aligned} x_{22} &= \left|13,22'\right|\sin\left(\alpha+\frac{\pi}{6}\right) = \begin{cases} R\sin\left(\dfrac{\theta}{2}\right) & \text{for } \sqrt{3}\tan\left(\dfrac{\theta}{2}\right) > \cos\eta \\ \dfrac{R}{2}\left(\sqrt{3}\cos\eta\cos\dfrac{\theta}{2}-\sin\dfrac{\theta}{2}\right) & \text{for } \sqrt{3}\tan\left(\dfrac{\theta}{2}\right) < \cos\eta \end{cases} \\ y_{22} &= \left|13,22'\right|\cos\left(\alpha+\frac{\pi}{6}\right) = \begin{cases} R\cos\eta\cos\left(\dfrac{\theta}{2}\right) & \text{for } \sqrt{3}\tan\left(\dfrac{\theta}{2}\right) > \cos\eta \\ \dfrac{R}{2}\left(\cos\eta\cos\dfrac{\theta}{2}+\sqrt{3}\sin\dfrac{\theta}{2}\right) & \text{for } \sqrt{3}\tan\left(\dfrac{\theta}{2}\right) < \cos\eta \end{cases} \\ z_{22} &= h = R\cos\left(\frac{\theta}{2}\right)\sin\eta \end{aligned} \tag{91}$$

Coordinates of point 24: $\Rightarrow$

$$x_{24} = x_{15} - \left(x_{22} - x_{13}\right) = \begin{cases} \sqrt{3}R\cos\eta\cos\dfrac{\theta}{2}, & \text{for } \sqrt{3}\tan\left(\dfrac{\theta}{2}\right) > \cos\eta \\ \dfrac{R}{2}\left(\sqrt{3}\cos\eta\cos\dfrac{\theta}{2} + 3\sin\dfrac{\theta}{2}\right), & \text{for } \sqrt{3}\tan\left(\dfrac{\theta}{2}\right) < \cos\eta \end{cases},$$

$$y_{24} = y_{22} = \begin{cases} R\cos\eta\cos\left(\dfrac{\theta}{2}\right), & \text{for } \sqrt{3}\tan\left(\dfrac{\theta}{2}\right) > \cos\eta \\ \dfrac{R}{2}\left(\cos\eta\cos\dfrac{\theta}{2} + \sqrt{3}\sin\dfrac{\theta}{2}\right), & \text{for } \sqrt{3}\tan\left(\dfrac{\theta}{2}\right) < \cos\eta \end{cases}, \tag{92}$$

$$z_{24} = h = R\cos\left(\frac{\theta}{2}\right)\sin\eta$$

Coordinates of point 23: $\Rightarrow$

$$\begin{aligned} x_{23} &= x_{14} = \frac{R}{2}\left(\sqrt{3}\cos\eta\cos\frac{\theta}{2} + \sin\frac{\theta}{2}\right), \\ y_{23} &= \frac{\sqrt{3}}{6}x_{15} = \frac{\sqrt{3}}{6}R\left(\sqrt{3}\cos\eta\cos\frac{\theta}{2} + \sin\frac{\theta}{2}\right), \\ z_{23} &= -\sqrt{\left(\frac{\sqrt{3}}{3}2R\right)^2 - x_{23}^2 - y_{23}^2} = -\frac{2}{\sqrt{3}}R\sqrt{1 - \frac{1}{4}\left(\sqrt{3}\cos\eta\cos\frac{\theta}{2} + \sin\frac{\theta}{2}\right)^2} \end{aligned} \tag{93}$$

Coordinates of point 4: $\Rightarrow$

$$\left|13,4'\right| = \sqrt{R^2 - H_1^2} = R\sqrt{1-\gamma_1^2} \tag{94}$$

$$\left|14',4'\right| = \sqrt{R^2 - \left(h + H_1\right)^2} = R\sqrt{1 - \left[\cos\left(\frac{\theta}{2}\right)\sin\eta + \gamma_1\right]^2} \tag{95}$$

$$\cos\beta_1 = \frac{\left|13,14'\right|^2 + \left|13,4'\right|^2 - \left|14',4'\right|^2}{2\left|13,14'\right|\left|13,4'\right|} = \frac{1 + 2\gamma_1\cos\left(\dfrac{\theta}{2}\right)\sin\eta}{2\sqrt{\sin^2\left(\dfrac{\theta}{2}\right)\sin^2\eta + \cos^2\eta}\sqrt{1-\gamma_1^2}} \tag{96}$$

$$\sin\beta_1 = \frac{\sqrt{3 - 4\left(\cos\dfrac{\theta}{2}\sin\eta + \gamma_1\right)^2 + 4\gamma_1\cos\dfrac{\theta}{2}\sin\eta}}{2\sqrt{\sin^2\left(\dfrac{\theta}{2}\right)\sin^2\eta + \cos^2\eta}\sqrt{1-\gamma_1^2}} \tag{97}$$

$$\Rightarrow \quad x_4 = |13,4'|\cos(\beta_1-\alpha) = R\frac{\left[\left(1+2\gamma_1\cos\left(\frac{\theta}{2}\right)\sin\eta\right)\left(\sqrt{3}\cos\eta\cos\frac{\theta}{2}+\sin\frac{\theta}{2}\right)+\sqrt{3-4\left(\cos\frac{\theta}{2}\sin\eta+\gamma_1\right)^2+4\gamma_1\cos\frac{\theta}{2}\sin\eta}\left(\cos\eta\cos\left(\frac{\theta}{2}\right)-\sqrt{3}\sin\left(\frac{\theta}{2}\right)\right)\right]}{4\left(\sin^2\left(\frac{\theta}{2}\right)\sin^2\eta+\cos^2\eta\right)}$$

$$y_4 = -|13,4'|\sin(\beta_1-\alpha) = -R\frac{\left[\sqrt{3-4\left(\cos\frac{\theta}{2}\sin\eta+\gamma_1\right)^2+4\gamma_1\cos\frac{\theta}{2}\sin\eta}\left(\sqrt{3}\cos\eta\cos\frac{\theta}{2}+\sin\frac{\theta}{2}\right)-\left(1+2\gamma_1\cos\left(\frac{\theta}{2}\right)\sin\eta\right)\left(\cos\eta\cos\left(\frac{\theta}{2}\right)-\sqrt{3}\sin\left(\frac{\theta}{2}\right)\right)\right]}{4\left[\sin^2\left(\frac{\theta}{2}\right)\sin^2\eta+\cos^2\eta\right]} \tag{98}$$

$$z_4 = -H_1 = -\gamma_1 R$$

Coordinates of point 6: $\Rightarrow$

On a similar note, the coordinates of point 6 can be obtained by replacing γ_1 with γ_2, and β_1 with β_2:

$$\begin{aligned} x_6 &= x_{15} - (x_4 - x_{13})\big|_{\substack{\gamma_2 \\ \beta_2}} \\ y_6 &= y_4\big|_{\substack{\gamma_2 \\ \beta_2}} \\ z_6 &= -H_2 \end{aligned} \tag{99}$$

$$a = \cos\frac{\theta}{2}\sin\eta,\ b = \cos\eta\cos\frac{\theta}{2},\ c = \sin\left(\frac{\theta}{2}\right)\sin\eta \tag{100}$$

$$d = (1+2\gamma_2 a),\ e = \sqrt{3-4(a+\gamma_2)^2+4\gamma_2 a} \tag{101}$$

$$\begin{aligned} x_6 &= R\left(\sqrt{3}b+\sin\frac{\theta}{2}\right) - R\frac{\left[d\left(\sqrt{3}b+\sin\frac{\theta}{2}\right)+e\left(b-\sqrt{3}\sin\left(\frac{\theta}{2}\right)\right)\right]}{4\left(c^2+\cos^2\eta\right)} \\ y_6 &= -R\frac{\left[e\left(\sqrt{3}b+\sin\frac{\theta}{2}\right)\right)-d\left(b-\sqrt{3}\sin\left(\frac{\theta}{2}\right)\right)\right]}{4\left[c^2+\cos^2\eta\right]} \\ z_6 &= -\gamma_2 R \end{aligned} \tag{102}$$

Coordinates of point 21: $\Rightarrow$

$$x_{21} = -|13,21'|\sin\left(\beta_2 - \alpha - \frac{\pi}{6}\right) = -R\frac{\left[\sqrt{3-4\left(\cos\frac{\theta}{2}\sin\eta+\gamma_2\right)^2+4\gamma_2\cos\frac{\theta}{2}\sin\eta}\left(\cos\eta\cos\frac{\theta}{2}+\sqrt{3}\sin\frac{\theta}{2}\right)\right]}{4\left[\sin^2\left(\frac{\theta}{2}\right)\sin^2\eta+\cos^2\eta\right]\sqrt{1-\gamma_2^2}}$$

$$\frac{-\left(1+2\gamma_2\cos\left(\frac{\theta}{2}\right)\sin\eta\right)\left(\sqrt{3}\cos\eta\cos\left(\frac{\theta}{2}\right)-\sin\left(\frac{\theta}{2}\right)\right)\Big]}{}$$

$$y_{21} = |13,21'|\cos\left(\beta_2 - \alpha - \frac{\pi}{6}\right) = R\frac{\left[\sqrt{3-4\left(\cos\frac{\theta}{2}\sin\eta+\gamma_2\right)^2+4\gamma_2\cos\frac{\theta}{2}\sin\eta}\left(\sqrt{3}\cos\eta\cos\frac{\theta}{2}-\sin\frac{\theta}{2}\right)\right]}{4\left[\sin^2\left(\frac{\theta}{2}\right)\sin^2\eta+\cos^2\eta\right]\sqrt{1-\gamma_2^2}}$$

$$\frac{+\left(1+2\gamma_2\cos\left(\frac{\theta}{2}\right)\sin\eta\right)\left(\cos\eta\cos\left(\frac{\theta}{2}\right)+\sqrt{3}\sin\left(\frac{\theta}{2}\right)\right)\Big]}{}$$

(103)

$$z_{21} = -H_2 = -\gamma_2 R$$

Dihedral angles α_1, α_2, and α_3*:*

The expressions for the three dihedral angles, α_1, α_2, and α_3 in terms of θ and η, are presented below,. α_1: $\Rightarrow$ First dihedral angle is between planes (13,14,23) and (13,22,23)

$$\alpha_1 = \arccos\left[\frac{\left(\frac{1}{2}R\right)^2+\left(\frac{1}{2}R\right)^2-|22,14|^2}{2\frac{1}{2}R\frac{1}{2}R}\right] = \arccos\left[1-8\sin^2\left(\frac{\theta}{2}\right)\right] \tag{104}$$

α_2: $\Rightarrow$ Second dihedral angle is between planes (13,14,23) and (15,14,23)

$$\alpha_2 = \angle(13,14,15) = \arccos\left[\frac{R^2+R^2-|13,15|^2}{2R^2}\right] = \arccos\left[1-\frac{1}{2}\left(\sqrt{3}\cos\eta\cos\frac{\theta}{2}+\sin\frac{\theta}{2}\right)^2\right] \tag{105}$$

α_3: $\Rightarrow$ Third dihedral angle is between planes (13,14,23) and (13,14,4). We need to find the coordinates of the mid-point between 13 and 23, say it is 100.

$$
\begin{aligned}
x_{100} &= \frac{1}{2}x_{23} = \frac{R}{4}\left(\sqrt{3}\cos\eta\cos\frac{\theta}{2}+\sin\frac{\theta}{2}\right),\\
y_{100} &= \frac{1}{2}y_{23} = \frac{\sqrt{3}}{12}R\left(\sqrt{3}\cos\eta\cos\frac{\theta}{2}+\sin\frac{\theta}{2}\right),\\
z_{100} &= \frac{1}{2}z_{23} = -\frac{1}{\sqrt{3}}R\sqrt{1-\frac{1}{4}\left(\sqrt{3}\cos\eta\cos\frac{\theta}{2}+\sin\frac{\theta}{2}\right)^2}
\end{aligned}
\tag{106}
$$

$$
\begin{aligned}
|4,100|^2 = R^2 &\left[\frac{\left[\left(1+2\gamma\cos\left(\frac{\theta}{2}\right)\sin\eta\right)\left(\sqrt{3}\cos\eta\cos\frac{\theta}{2}+\sin\frac{\theta}{2}\right)\right.}{\left.+\sqrt{3-4\left(\cos\frac{\theta}{2}\sin\eta+\gamma\right)^2+4\gamma\cos\frac{\theta}{2}\sin\eta}\left(\cos\eta\cos\left(\frac{\theta}{2}\right)-\sqrt{3}\sin\left(\frac{\theta}{2}\right)\right)\right]}{4\left(\sin^2\left(\frac{\theta}{2}\right)\sin^2\eta+\cos^2\eta\right)}-\frac{1}{4}\left(\sqrt{3}\cos\eta\cos\frac{\theta}{2}+\sin\frac{\theta}{2}\right)\right]^2\\
+R^2 &\left[\frac{\left[\sqrt{3-4\left(\cos\frac{\theta}{2}\sin\eta+\gamma\right)^2+4\gamma\cos\frac{\theta}{2}\sin\eta}\left(\sqrt{3}\cos\eta\cos\frac{\theta}{2}+\sin\frac{\theta}{2}\right)\right.}{\left.-\left(1+2\gamma\cos\left(\frac{\theta}{2}\right)\sin\eta\right)\left(\cos\eta\cos\left(\frac{\theta}{2}\right)-\sqrt{3}\sin\left(\frac{\theta}{2}\right)\right)\right]}{4\left[\sin^2\left(\frac{\theta}{2}\right)\sin^2\eta+\cos^2\eta\right]}-\frac{\sqrt{3}}{12}\left(\sqrt{3}\cos\eta\cos\frac{\theta}{2}+\sin\frac{\theta}{2}\right)\right]^2\\
+R^2 &\left[-\gamma+\frac{1}{\sqrt{3}}\sqrt{1-\frac{1}{4}\left(\sqrt{3}\cos\eta\cos\frac{\theta}{2}+\sin\frac{\theta}{2}\right)^2}\right]^2\\
&= R^2D
\end{aligned}
\tag{107}
$$

$$
\alpha_3 = \arccos\left[\frac{\left(\frac{R}{2\sqrt{3}}\right)^2+\left(\frac{\sqrt{3}}{2}R\right)^2-R^2D}{2\frac{R}{2\sqrt{3}}\frac{\sqrt{3}}{2}R}\right] = \arccos\left[\frac{5}{3}-2D\right] \tag{108}
$$

Derivatives of Dihedral Angles

In this section, the relationship between the nine dihedral angles (see Table 4.4) and dihedral angles used in ABAQUS UEL code will found. Subsequently, the first and the second of the nine angles will be found.

Angle #1: $\angle(22,23,14)$

$$\angle(22,23,14)=\arccos\left[\frac{\left(\frac{R}{\sqrt{3}}\right)^2+\left(\frac{R}{\sqrt{3}}\right)^2-|22,14|^2}{2\frac{R}{\sqrt{3}}\frac{R}{\sqrt{3}}}\right] \tag{109}$$

$$=\arccos\left[1-6\sin^2\left(\frac{\theta}{2}\right)\right]=\arccos\left(\frac{1}{4}+\frac{3}{4}\cos\alpha_1\right) \tag{110}$$

$$\alpha_1=\arccos\left[\frac{4}{3}\cos\angle(22,23,14)-\frac{1}{3}\right] \tag{111}$$

$$\frac{\partial\alpha_1}{\partial\angle(22,23,14)}=\frac{4}{3}\frac{\sin\angle(22,23,14)}{\sin\alpha_1} \tag{112}$$

$$\frac{\partial^2\alpha_1}{\partial^2\angle(22,23,14)}=\frac{4}{3}\left[\frac{\cos\angle(22,23,14)}{\sin\alpha_1}-\frac{\sin\angle(22,23,14)}{\sin^2\alpha_1}\cos\alpha_1\frac{\partial\alpha_1}{\partial\angle(22,23,14)}\right] \tag{113}$$

Angle #2: $\angle(13,23,15)$

$$\angle(13,23,15)=\arccos\left[\frac{\left(\frac{2R}{\sqrt{3}}\right)^2+\left(\frac{2R}{\sqrt{3}}\right)^2-|13,15|^2}{2\frac{2R}{\sqrt{3}}\frac{2R}{\sqrt{3}}}\right] \tag{114}$$

$$=\arccos\left[1-\frac{3}{8}\left(\sqrt{3}\cos\eta\cos\frac{\theta}{2}+\sin\frac{\theta}{2}\right)^2\right]=\arccos\left(\frac{1}{4}+\frac{3}{4}\cos\alpha_2\right)$$

$$\alpha_2=\arccos\left[\frac{4}{3}\cos\angle(13,23,15)-\frac{1}{3}\right] \tag{115}$$

$$\frac{\partial\alpha_2}{\partial\angle(13,23,15)}=\frac{4}{3}\frac{\sin\angle(13,23,15)}{\sin\alpha_2} \tag{116}$$

$$\frac{\partial^2\alpha_2}{\partial^2\angle(13,23,15)}=\frac{4}{3}\left[\frac{\cos\angle(13,23,15)}{\sin\alpha_2}-\frac{\sin\angle(13,23,15)}{\sin^2\alpha_2}\cos\alpha_2\frac{\partial\alpha_2}{\partial\angle(13,23,15)}\right] \tag{117}$$

Angle #3: $\angle(13,23,24)$

We can set this angle's stiffness as zero and bypass it!

Angle #4: $\angle(23,14,4)$

$$|4,23|^2 = \left(\frac{R}{\sqrt{3}}\right)^2 + R^2 - 2\frac{R}{\sqrt{3}}R\cos\angle(23,14,4) \tag{118}$$

Also,

$$|4,23|^2 = \left(\frac{2R}{\sqrt{3}}\right)^2 + R^2 - 2\frac{2R}{\sqrt{3}}R\cos\angle(23,13,4)\,. \tag{119}$$

So,

$$\cos\angle(23,13,4) = \frac{\sqrt{3}}{4} + \frac{1}{2}\cos\angle(23,14,4) \tag{120}$$

$$\begin{aligned} |4,100|^2 &= \left(\frac{R}{\sqrt{3}}\right)^2 + R^2 - 2\frac{R}{\sqrt{3}}R\cos\angle(23,13,4) \\ &= \frac{4}{3}R^2 - \frac{2}{\sqrt{3}}R^2\left[\frac{\sqrt{3}}{4} + \frac{1}{2}\cos\angle(23,14,4)\right] \\ &= \frac{5}{6}R^2 - \frac{R^2}{\sqrt{3}}\cos\angle(23,14,4) \end{aligned} \tag{121}$$

$$\alpha_3 = \arccos\left[\frac{\left(\frac{R}{2\sqrt{3}}\right)^2 + \left(\frac{\sqrt{3}}{2}R\right)^2 - |4,100|^2}{2\frac{R}{2\sqrt{3}}\frac{\sqrt{3}}{2}R}\right] = \arccos\left[\frac{2}{\sqrt{3}}\cos\angle(23,14,4)\right] \tag{122}$$

$$\frac{\partial\alpha_3}{\partial\angle(23,14,4)} = \frac{2}{\sqrt{3}}\frac{\sin\angle(23,14,4)}{\sin\alpha_3} \tag{123}$$

$$\frac{\partial^2\alpha_3}{\partial^2\angle(23,14,4)} = \frac{2}{\sqrt{3}}\left[\frac{\cos\angle(23,14,4)}{\sin\alpha_3} - \frac{\sin\angle(23,14,4)}{\sin^2\alpha_3}\cos\alpha_3\frac{\partial\alpha_3}{\partial\angle(23,14,4)}\right]. \tag{124}$$

Angle #5: We can set this angle's stiffness as zero and bypass it in the code.

Angle #6: $\angle(23,13,4)$.

$$\alpha_3 = \arccos\left[\frac{2}{\sqrt{3}}\cos\angle(23,14,4)\right] = \arccos\left[-1+\frac{4}{\sqrt{3}}\cos\angle(23,13,4)\right] \tag{125}$$

$$\frac{\partial\alpha_3}{\partial\angle(23,13,4)} = \frac{4}{\sqrt{3}}\frac{\sin\angle(23,13,4)}{\sin\alpha_3} \tag{126}$$

$$\frac{\partial^2\alpha_3}{\partial^2\angle(23,13,4)} = \frac{4}{\sqrt{3}}\left[\frac{\cos\angle(23,13,4)}{\sin\alpha_3} - \frac{\sin\angle(23,13,4)}{\sin^2\alpha_3}\cos\alpha_3\frac{\partial\alpha_3}{\partial\angle(23,13,4)}\right] \tag{127}$$

Angle #7: $\angle(13,14,15)$. It is just the second dihedral angle α_2.

$$\alpha_2 = \angle(13,14,15) = \arccos\left[\cos\angle(13,14,15)\right] \tag{128}$$

Angle #8: We can set this angle's stiffness is zero and by pass it in the code.

Angle #9: $\angle(22,13,14)$ here or $\angle(6,1,7)$ in the ppt file.

$$\alpha_1 = \arccos\left[1-8\sin^2\left(\frac{\angle(22,13,14)}{2}\right)\right] = \arccos\left[4\cos\angle(22,13,14)-3\right] \tag{129}$$

$$\frac{\partial\alpha_1}{\partial\angle(22,13,14)} = 4\frac{\sin\angle(22,13,14)}{\sin\alpha_1} \tag{130}$$

$$\frac{\partial^2\alpha_1}{\partial^2\angle(22,13,4)} = 4\left[\frac{\cos\angle(22,13,4)}{\sin\alpha_1} - \frac{\sin\angle(22,13,4)}{\sin^2\alpha_1}\cos\alpha_1\frac{\partial\alpha_1}{\partial\angle(22,13,4)}\right] \tag{131}$$

Summary:

Table 4.6: Summary of Ron Resch Type I Element

Angle type	*Angle*	*Dihedral angle*	*Stiffness*
1	(2,1,3), (2,1,4), (3,1,4)	α_1	T_1
2	(5,1,6), (5,1,7), (6,1,7)	α_2	T_2
3	Bypass	NA	NA
4	(1,2,8), (1,2,10), (1,3,11), (1,3,13), (1,4,14), (1,4,16)	α_3	T_3
5	Bypass	NA	NA
6	Bypass	NA	NA

Table 4.7: Summary of Ron Resch Type II Element

Angle type	*Angle*	*Dihedral angle*	*Stiffness*
1	(1,2,8), (1,2,10)	α_1	T_1
2	(1,3,11), (1,3,13)	α_2	T_2
3	Bypass	NA	NA
4	(2,1,6), (2,1,7), (1,4,15), (1,5,17)	α_3	T_3
5	Bypass	NA	NA
6	Bypass	NA	NA
7	Bypass	NA	NA
8	Bypass	NA	NA
9	Bypass	NA	NA

The Table 4.6 and Table 4.7 show which stiffnesses have to be considered while evaluating the energy contributions by type I and type II elements, respectively, to avoid over counting of energy.

4.2.3 STEP 3: WRITING THE UEL CODE FOR RR

The UEL code is attached in the appendix. It has four subroutines and thirteen functions. A brief description of the three subroutines is given below.,

1. SUBROUTINE UEL: This is a main subroutine called by ABAQUS while running the program. This will be called for each element that is of a general user-defined element type (i.e., not defined by a linear stiffness or mass matrix read either directly or from results file data) each time element calculations are required and must perform all of the calculations for the element, appropriate to the current activity in the analysis.

2. SUBROUTINE CALEMEF: This subroutine will be called by SUBROUTINE UEL to evaluate the 57x57 elemental stiffness matrix, and a 57x1 non-equilibrium force vector for each element.

3. DUBROUTINE SPRINGEMEF: This subroutine will be called by SUBROUTINE UEL to evaluate the 6x6 spring stiffness matrix, and a 6x1 non-equilibrium force vector for each spring element.

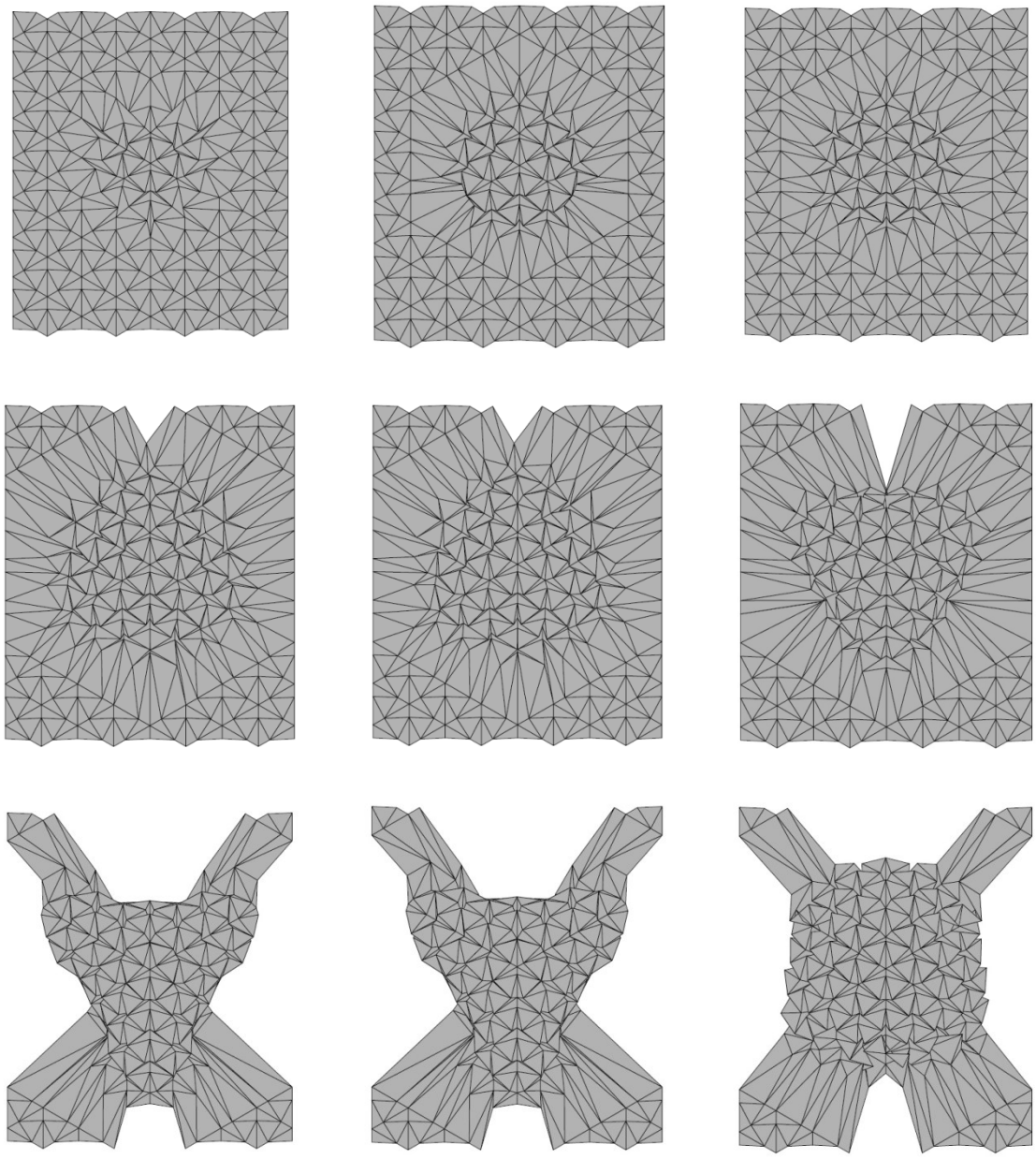
4. SUBROUTINE DIFF: This subroutine will be called by SUBROUTINE CALEMEF to evaluate the first and second derivate of the total energy. This is the corner stone of the entire program, as a lot of math is needed to be consistent in order for this phase to work. The relationships derived in step 2 will be used in this part to evaluate the partial derivatives of the total energy using the chain rule.

4.3 Obtaining the Equilibrium Configuration

An equilibrium configuration of a system is a configuration corresponding to its minimum energy when subjected to a particular boundary condition. The equilibrium configuration most likely changes when the boundary conditions are altered. It is very important to start an FEA analysis of a system from its equilibrium configuration. Otherwise, the system will be constantly under the influence of unknown random external forces, which could make the solution oscillate, making it practically impossible to obtain a meaningful solution.

When dealing with a periodic origami such as MO, it is easy to obtain a closed form solution for most of the unknown quantities. So the coordinates of the nodes or the initial configuration is very close to the equilibrium solution. As a result, one need not carry out an initial run to obtain the equilibrium configuration. Instead, simulations can be directly carried out on the initial configuration. On the other hand, in the case of RR, the coordinates are obtained by solving many equations. These values might be in the local equilibrium zones, but collectively, they could be far away from the global equilibrium. So, a step by step procedure is used to drag all the nodes into the global equilibrium zone by enforcing suitable (fictitious) boundary conditions. Once all the nodes are in the proper zone, the boundary conditions are released and a final equilibrium configuration is obtained.

During this procedure, a tier by tier approach is used. First a single unit cell is selected, and its nodes are manually moved in to make it converge to an equilibrium solution. Then, the converged coordinates are fixed, and the closest tier unit cells surrounding the center unit cells are introduced into the solution in a subsequent step. Once this system converges, the next tier is introduced, and so on (see Figure 4.8). This method has very few degrees of freedom when compared to the one where all the nodes are introduced simultaneously. Hence, cost is less and faster convergence is imminent.



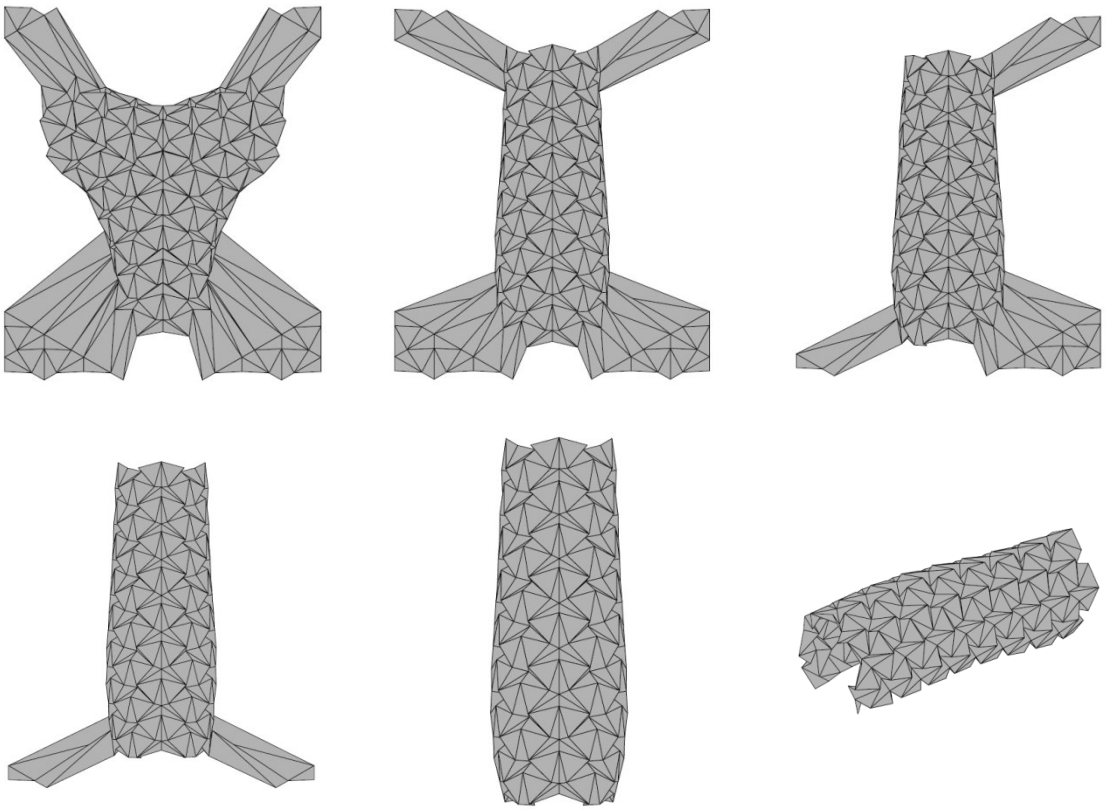


Figure 4.8: Evolution of Ron Resch crease pattern

This procedure is very tedious and time consuming. A lot of effort can be avoided by using an initial configuration closer to the equilibrium configuration or, at least, a configuration in which the nodes are positioned in such a way that all nine dihedral angles are in the proper zone. Some of the dihedral angles have more than one solution; and it is very hard to impose constraints on the system to accurately obtain a unique solution. These different solutions are numerically separated by barriers which are numerically impossible to overcome. So, it is very important to push the system in the right direction by making sure that all the angles are in the right zone.

CHAPTER 5 : RESULTS

In this section, the results of some basic simulations for RR and MO will be shown and discussed.

5.1 MIURA-ORI

This is a periodic origami, and it takes relatively shorter times to reach convergence. The UEL code for this element type is presented in APPENDIX I, and the input files for the simulation in this section can be found in APPENDIX II.

5.1.1. MODAL ANALYSIS

Using the MO UEL, a modal analysis was performed on a 26X26 MO pattern. This pattern has 26 nodes along X direction and 26 nodes along the Y direction. For this analysis, all the degrees of freedom of the left most array (containing the nodes 1, 27… 651) are confined (shown in Figure 5.1). The first four mode shapes are shown below in Figure 5.2

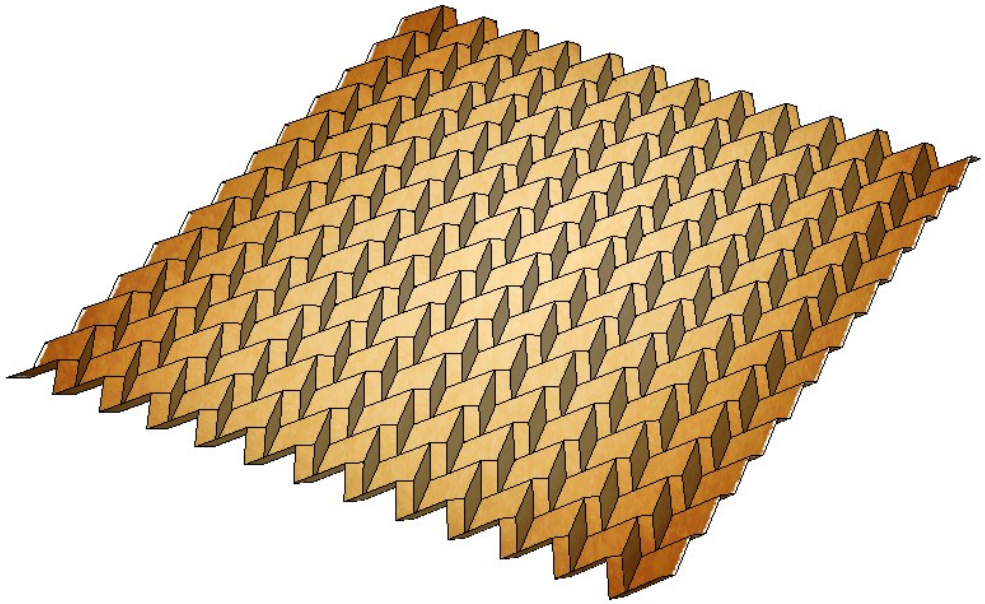
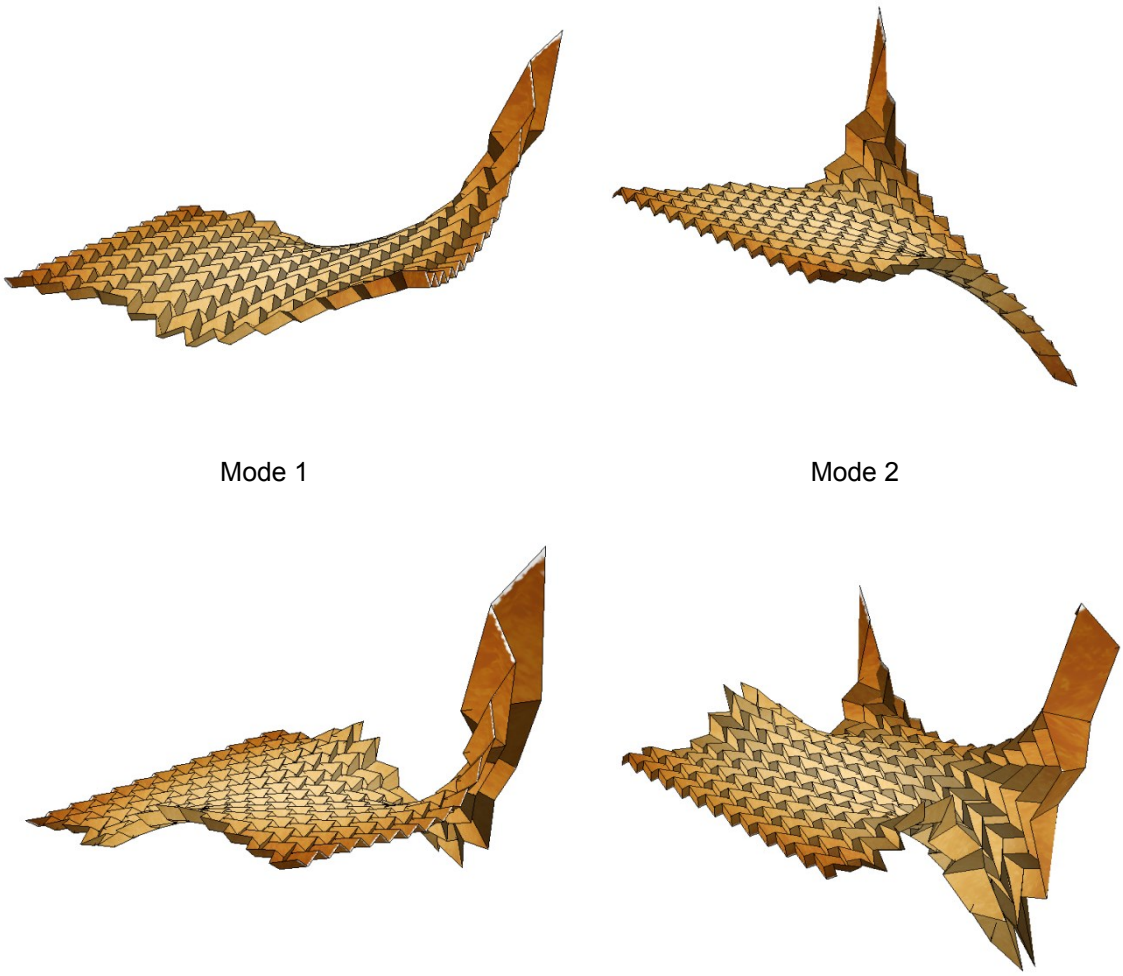


Figure 5.1: Equilibrium configuration of 26X26 Miura-Ori pattern



Mode 1 Mode 2

Mode 3 Mode 4

Figure 5.2: First Four Modes of Miura-Ori

5.1.2. SHEAR ANALYSIS

In this analysis, a 13X13 MO pattern was used. All the degrees of freedom of the bottom most array (containing the nodes 1, 2…13) were confined. The result of the simulation can be seen in the Figure 5.3. The red lines show the undeformed configuration which is superimposed over the deformed configuration.

The interesting thing to note here is that the right-hand-side of the structure tends to deformations contradictory to the direction of applied force. This shows that the structure has negative bulk modulus. The in-depth analysis is reserved for future works.

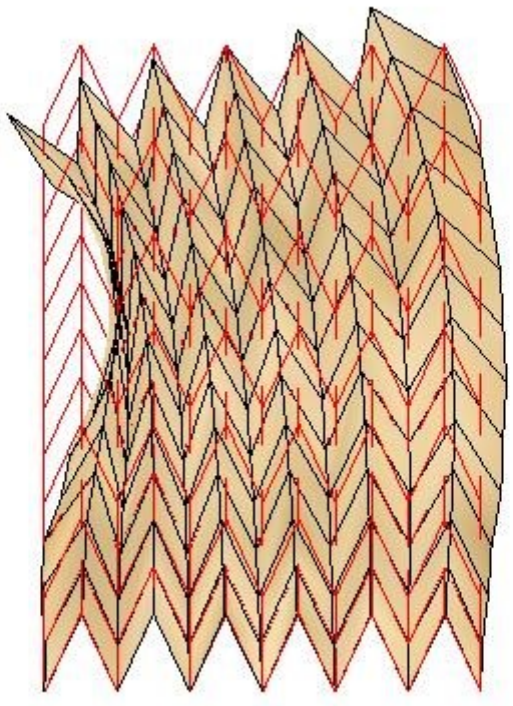


Figure 5.3: Deformed Miura-Ori Subjected to Shear Force

5.1.3. TORSION AND BENDING ANALYSIS

It can be seen from Figure 5.2 that the first two failure modes could result from bending and torsion. So, a bending and torsion analysis was conducted on a 13X13 MO pattern. Figure 5.4 shows the deformed configuration due to a torsion force, and Figure 5.5 shows the deformed configuration due to a bending force.

Jiang et al developed a Lithium Ion battery that folds up like an MO pattern. The following two analyses were used to prove that the battery sustains minimum strain due to torsion and bending forces. It must be noted that the battery is extremely thin, so it is feasible to use this element type in the FEA formulation.

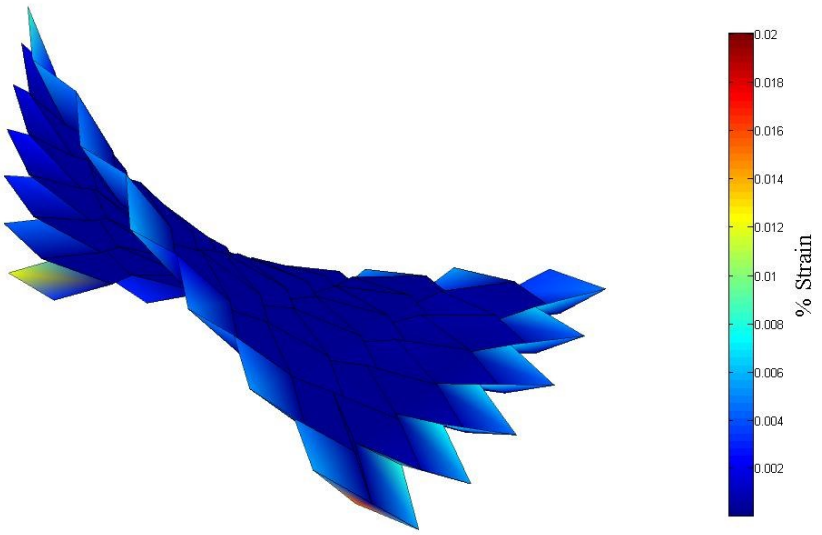


Figure 5.4: Deformed Miura-Ori subjected to Torsion force

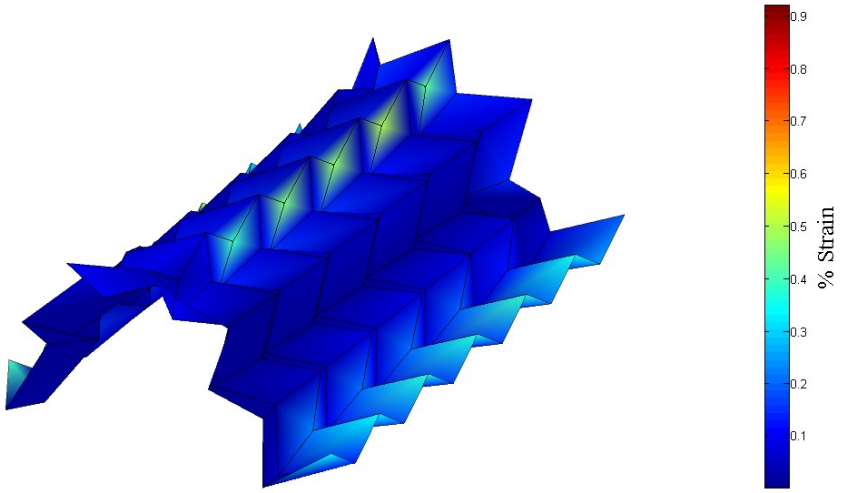


Figure 5.5: Deformed Miura-Ori Subjected to Bending Force

It has already been stated that a rigid origami experiences very minimal strain on the faces. To prove this statement, a strain contour map is superimposed over the deformed configuration to get an idea of the magnitude of strain at various places in the pattern. And it can be seen from the above figure that the strain is indeed minimum.

5.2 RON RESCH

This is a non-periodic origami, and it takes relative longer times to reach convergence. This is because the equilibrium configuration is unknown. First, the equilibrium configuration is attained, and then the required analysis is conducted on it. The UEL code for this element type is presented in APPENDIX I, and the input files for the simulation in this section can be found in APPENDIX II.

5.2.1. BUCKLING

Using the procedure discussed in the section 4.3, the equilibrium configuration was obtained. Then a compressive load was applied on the right-hand side while fixing the left-hand side. This test was used to determine if the structure buckles. The results are presented below.

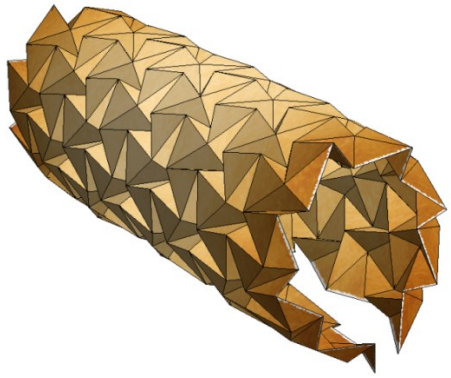


Figure 5.6: Eq. Configuration

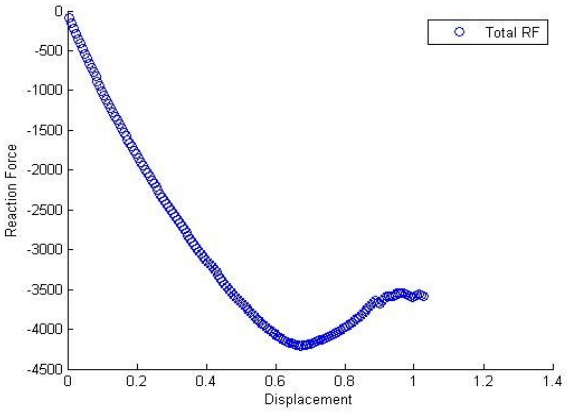


Figure 5.7: Disp. vs. RF Curve

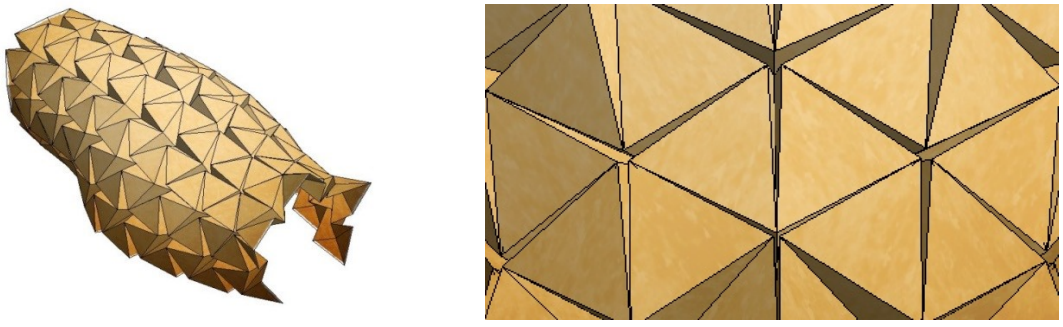
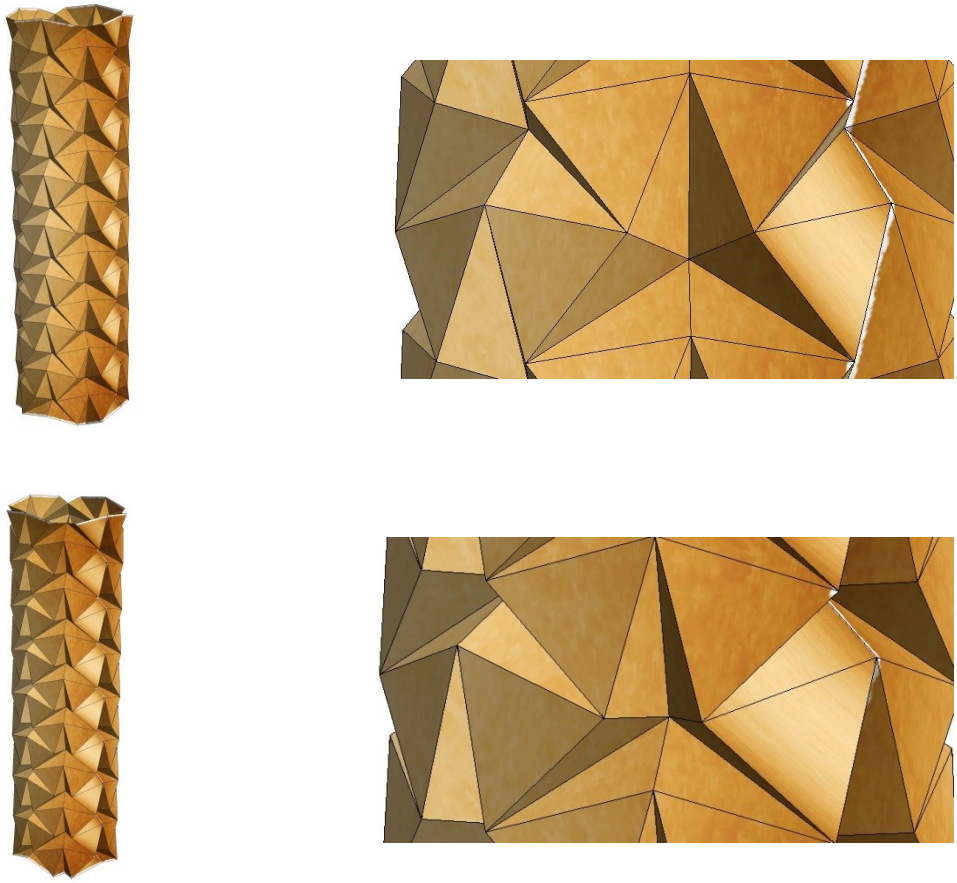


Figure 5.8: Buckled Structure and Interlocked Faces

Figure 5.6 shows the equilibrium configuration; Figure 5.7 shows the buckled shape and interlocked faces; and Figure 5.8 shows displacement vs. reaction curve. It can be observed that magnitude of reaction force increases, then suddenly drops, then retains a specific value and again increases. The sudden drop in reaction force corresponds to the point where the structure buckles. Due to interlocking action between the various faces, the reaction force starts to increase again.

5.2.2. RRT COMPRESSION

After observing the interlocking mechanism in section 5.2.1, it was determined to achieve this over the entire circumference rather than at a single location. Achieving this could result in increased resistance to buckling for tubular structures. The boundary conditions used in section 5.2.1 were reproduced on a tube (see Figure 5.9). The results of the simulation are shown below.



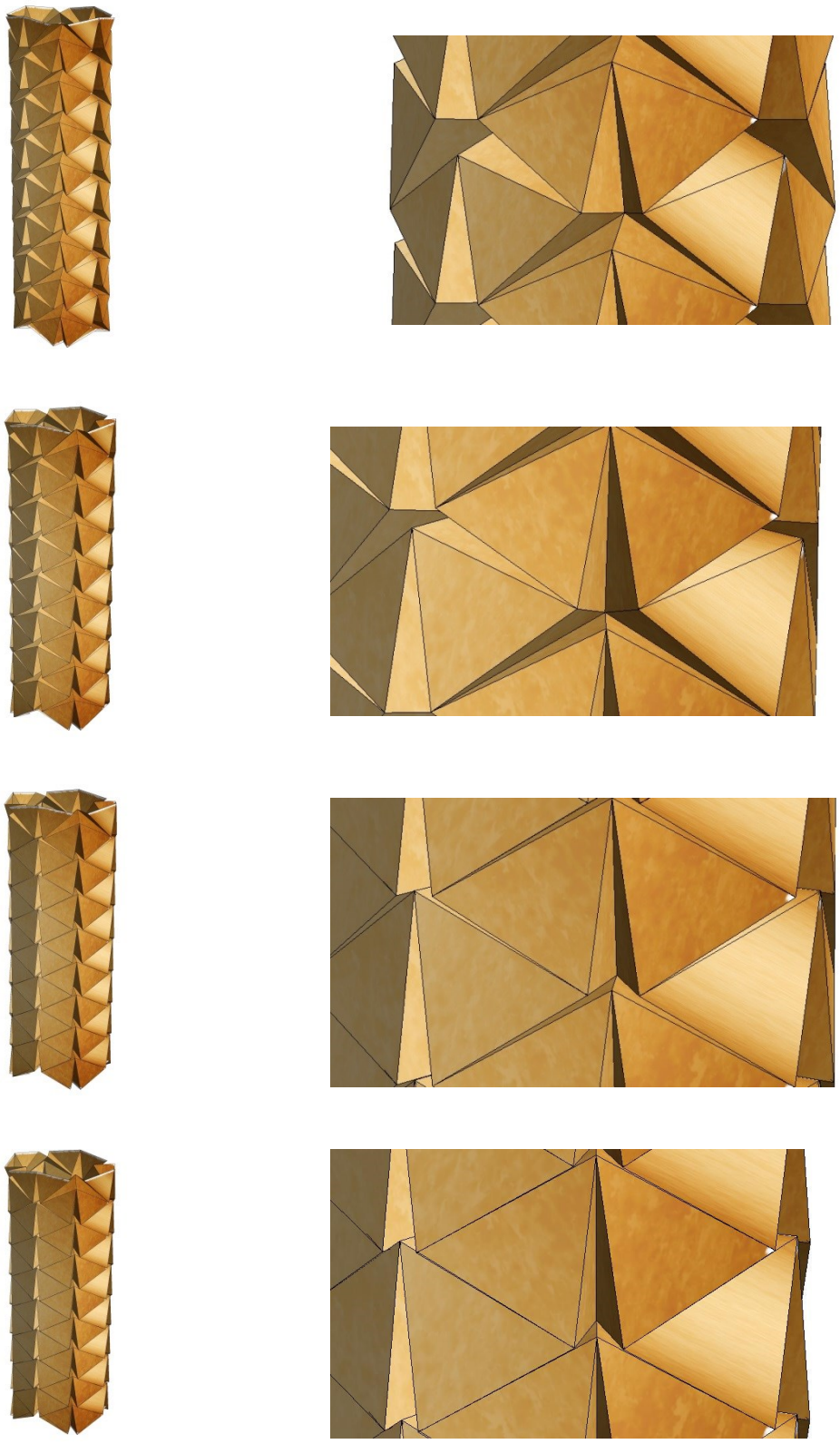


Figure 5.9: Ron Resch compression analysis

The last structure in the Figure 5.9 has very high resistance to buckling due to the interlocking of all neighboring faces.

5.3 FUTURE WORK

RO have many unique properties that result from their unique way of folding. For the periodic Miura-Ori, this analytical approach can be used to study both its negative and positive in-plane Poisson's ratio, and analyze how they reflect on the various moduli of the structure. It was also observed that the RR has exceptionally high loading bearing capability. Using this model, the buckling stability of Ron Resch tubes can be analyzed. The approach in this paper paves the way for the study of interesting and unique geometric and mechanical properties of origami structures as mechanical metamaterials.

REFERENCES

1. J. P. Gardner et al., Space Science Reviews 123, 485 (2006).

2. C. Cromvik, K. Eriksson, Airbag Folding Based on Origami Mathematics (Chalmers University of Technology, 2006),

3. K. Kuribayashi et al., Materials Science and Engineering a-Structural Materials Properties Microstructure and Processing 419, 131 (Mar, 2006).

4. R. Tang et al., Applied Physics Letters 104, 083501 (2014).

5. Z. Song et al., Nature Communications 5:3140 doi:10.1038/ncomms4140 (2014).

6. R. J. Lang, in Proceeding SCG '96 Proceedings of the twelfth annual symposium on computational geometry. (1996) pp. 98-105.

7. R. J. Lang, Origami Design Secrets (CRC Press ed. 2nd, 2011

8. H. C. Greenberg, M. L. Gong, S. P. Magleby, L. L. Howell, Mechanical Science 2, 217 (2011).

9. E. Cerda, L. Mahadevan, Proceedings of the Royal Society a-Mathematical Physical and Engineering Sciences 461, 671 (Mar, 2005).

10. M. A. Dias, C. D. Santangelo, Europhysics Letters100 (2012).

11. M. A. Dias, L. H. Dudte, L. Mahadevan, C. D. Santangelo, Physical Review Letters109, 5 (Sep, 2012).

12. Z. Y. Wei, Z. V. Guo, L. Dudte, H. Y. Liang, L. Mahadevan, Physical Review Letters110 (May, 2013).

13. S.-M. Belcastro, T. C. Hull, Linear Algebra and its Applications 348, 273 (2002).

14. S.-M. Belcastro, T. C. Hull, in Origami 3 T. C. Hull, Ed. (A K Peters/CRC Press, 2002) pp. 39-51.

15. T. Tachi, in Origami 4, Proceedings of 4 OSME: 4th International Conference on Origami Science, Mathematics and Education R. J. Lang, Ed. (2009) pp. 175-188.

16. K. Miura, “Method of packaging and deployment of large membranes in space.” (Institute of Space and Astronomical Sciences, 1985).

17. M. Schenk, S. D. Guest, Proceedings of the National Academy of Sciences of the United States of America 110, 3276 (Feb, 2013).

18. R. D. Resch, U. S. P. Office, Ed. (USA, 1968), vol. 3,407,558.

19. B. Liu, Y. Huang, H. Jiang, S. Qu, K. C. Hwang, Computer Methods in Applied Mechanics and Engineering 193, 1849 (2004).

20. B. Liu et al., Physical Review B 72, 8 (Jul, 2005).

21. K.J. Bathe and E.L. Wilson, “NONSAP—A Nonlinear Structural Analysis Program”, J. Nuclear Engineering and Design, 29, 266–293, 1974.

22. K.J. Bathe and E.L. Wilson, “Solution Methods for Eigenvalue Problems in Structural Mechanics”, Int. J. for Numerical Methods in Engineering, 6, 213–266, 1973.

APPENDIX I

UEL CODES FOR NFEM ELEMENT TYPES

UEL FOR MO ELEMENT TYPE

```
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .          U E L   F O R   P E R I O D I C   O R I G A M I           .
C .               M I U R A   O R I   P A T T E R N                    .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

      SUBROUTINE UEL(RHS,AMATRX,SVARS,ENERGY,NDOFEL,NRHS,NSVARS,
     1 PROPS,NPROPS,COORDS,MCRD,NNODE,U,DU,V,A,JTYPE,TIME,DTIME,
     2 KSTEP,KINC,JELEM,PARAMS,NDLOAD,JDLTYP,ADLMAG,PREDEF,NPREDF,
     3 LFLAGS,MLVARX,DDLMAG,MDLOAD,PNEWDT,JPROPS,NJPROP,PERIOD)

      INCLUDE 'ABA_PARAM.INC'
      REAL*8 KB
      PARAMETER ( NTOTPART = 1033, NTOTEL = 1033)
      PARAMETER ( SHIFT = 0.0 )

      DIMENSION RHS(MLVARX,*),AMATRX(NDOFEL,NDOFEL),PROPS(*),
     1 SVARS(NSVARS),ENERGY(8),COORDS(MCRD,NNODE),U(NDOFEL),
     2 DU(MLVARX,*),V(NDOFEL),A(NDOFEL),TIME(2),PARAMS(*),
     3 JDLTYP(MDLOAD,*),ADLMAG(MDLOAD,*),DDLMAG(MDLOAD,*),
     4 PREDEF(2,NPREDF,NNODE),LFLAGS(*),JPROPS(*)

      REAL*8 EX(9,3),EK(27,27),EF(27)
      REAL*8 SPRINGEX(2,3),SPRINGEX0(2,3),SPRINGEK(6,6),SPRINGEF(6)
      INTEGER NELE(9)
      COMMON ITIME
      COMMON ETTOT,ESPR
      REAL*8 MASS

      MASS = 1.D0
      OPEN(2000, FILE = 'C:\Users\Deepak\Desktop\cheng\coors_MO.TXT')
      OPEN(105, FILE = 'C:\Users\Deepak\Desktop\cheng\energy_MO.dat')
      OPEN(6000, FILE = 'C:\Users\Deepak\Desktop\cheng\warning_MO.dat')

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C . E L E M E N T   A R R A Y   S E Q U E N C E                        .
C .                                                                    .
C .                                                                    .
C . NELE(1) = 1 (always)                                               .
C . NELE(2) = 2                                                        .
C . NELE(3) = 3                                                        .
C . NELE(4) = 4                                                        .
C . NELE(5) = 5                                                        .
C . NELE(6) = 6                                                        .
C . NELE(7) = 7                                                        .
C . NELE(8) = 8                                                        .
C . NELE(9) = 9                                                        .
C . IF NELE(I) = 1, FOR I = 2 TO 9, IT MEANS THAT THE NODE DOES NOT    .
C . EXIST                                                              .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

C
```

```
C  RESET THE TOTAL ENERGY IN ALL THE ELEMENTS TO ZERO AFTER EVERY
C  ITERATION
C
      IF(ITIME.EQ.0) THEN
         ETTOT = 0.D0
         ESPR = 0.D0
      ENDIF

      AMATRX = 0.D0

      DO K1 = 1,NDOFEL
         DO KRHS = 1,NRHS
            RHS(K1,KRHS) = 0.D0
          ENDDO
      ENDDO
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C . E L E M E N T  T Y P E S                                           .
C .                                                                    .
C . JTYPE = 1: 9 NODES, 6 DIHEDERAL ANGLES                             .
C . JTYPE = 2: 2 NODES, LENGTH = 1                                     .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

C
C  CALCULATING THE CURRENT COORDINATES
C
      IF(JTYPE.EQ.1) THEN
             DO INODE = 1,NNODE
                DO IMCRD = 1,MCRD
                     IPOSN = 3*(INODE-1) + IMCRD
                   EX(INODE,IMCRD) = COORDS(IMCRD,INODE) + U(IPOSN)
                ENDDO
             ENDDO
         NELE = 1
C
C  CHECK IF NODE EXISTS, I.E., IF ITS DISTANCE FROM CENTRAL NODE IS
C  GREATER THAN 0
C
         DO INODE = 2,NNODE
            DIS = DSQRT((EX(INODE,1)-EX(1,1))**2.D0
     &       + (EX(INODE,2)-EX(1,2))**2.D0 + (EX(INODE,3)-
EX(1,3))**2.D0)
            IF(DIS.LT.1.E-7) THEN
                   NELE(INODE) = 1
                ELSE
                   NELE(INODE) = INODE
                ENDIF
         ENDDO
C
C  CALLING CALEMEF TO OBTAIN ELEMENTAL STIFFNESS AND FORCE VECTOR

C
             CALL CALEMEF(EX,NELE,EK,EF,ETTOT,JELEM)
      ENDIF

      IF(JTYPE.EQ.2) THEN
```

```
              DO INODE = 1,NNODE
                   DO IMCRD = 1,MCRD
                      IPOSN = IMCRD + 3*(INODE-1)
                      SPRINGEX(INODE,IMCRD) = COORDS(IMCRD,INODE) +
U(IPOSN)
                      SPRINGEX0(INODE,IMCRD) = COORDS(IMCRD,INODE)
                   ENDDO
              ENDDO
C
C  CALLING SPRINGEMEF TO OBTAIN SPRING STIFFNESS AND FORCE VECTOR
C
              CALL SPRINGEMEF(SPRINGEX,SPRINGEX0,SPRINGEK,SPRINGEF,ESPR,
     &                        JELEM)
       ENDIF

       IF(LFLAGS(1).EQ.1 .OR. LFLAGS(1).EQ.2) THEN
              IF(JTYPE.EQ.1) THEN
                 RHS(1,1) = -EF(1)
                   RHS(2,1) = -EF(2)
                   RHS(3,1) = -EF(3)
                   DO K1 = 1,NDOFEL
                      DO K2 = 1,NDOFEL
                           AMATRX(K2,K1) = EK(K2,K1)
                      ENDDO
                 ENDDO
C
C  A CONSTANT IS ADDED TO THE DIAGONAL TERMS TO MAKE THE AMATRIX
C  POSITIVE DEFINTIE
C
                   AMATRX(1,1) = AMATRX(1,1) + 0D1
                   AMATRX(2,2) = AMATRX(2,2) + 0D1
                   AMATRX(3,3) = AMATRX(3,3) + 0D1
                   WRITE(2000,2000) JELEM,EX(1,1),EX(1,2),EX(1,3)
2000               FORMAT(I8,',',f20.15,',',f20.15,',',f20.15)
              ENDIF

              IF(JTYPE.EQ.2) THEN
                   RHS(1,1) = -SPRINGEF(1)
                   RHS(2,1) = -SPRINGEF(2)
                   RHS(3,1) = -SPRINGEF(3)
                   RHS(4,1) = -SPRINGEF(4)
                   RHS(5,1) = -SPRINGEF(5)
                   RHS(6,1) = -SPRINGEF(6)
                   DO K1 = 1,NDOFEL
                      DO K2 = 1,NDOFEL
                           AMATRX(K2,K1) = SPRINGEK(K2,K1)
                      ENDDO
                 ENDDO
              ENDIF
       ENDIF
C
C  THE VALUE OF LFLAGS DETERMINES THE TYPE OF SIMULATION
C
       IF(LFLAGS(1).EQ.41) THEN
              IF(LFLAGS(3).EQ.2) THEN
                   IF(JTYPE.EQ.1) THEN
                      DO K1 = 1,NDOFEL
```

```
                        DO K2 = 1,NDOFEL
                             AMATRX(K2,K1) = EK(K2,K1)
                        ENDDO
                    ENDDO
                ENDIF
                IF(JTYPE.EQ.2) THEN
                    DO K1 = 1,NDOFEL
                       DO K2 = 1,NDOFEL
                            AMATRX(K2,K1) = SPRINGEK(K2,K1)
                        ENDDO
                    ENDDO
                ENDIF
            ENDIF
            IF(LFLAGS(3).EQ.4) THEN
                    IF(JTYPE.EQ.1) THEN
                       DO K1 = 1,3
                          AMATRX(K1,K1) = MASS
                       ENDDO
                    ENDIF
                    IF(JTYPE.EQ.2) THEN
                       DO K1 = 1,NDOFEL
                          AMATRX(K1,K1) = MASS
                       ENDDO
                    ENDIF
                ENDIF
        ENDIF

        ITIME = ITIME + 1
C
C  PORT NUMBER 7 IS USED TO WRITE DATA INTO MSG FILE
C  USED ONLY FOR DEBUG
C  FOR DEBUG PURPOSE
C        WRITE(7,*) 'ICALL=',ITIME
C  WRTING AFTER EVERY ITERATION, I.E., ITIME.EQ.NTOTEL
C  AND REWINDING
C
        IF(ITIME.EQ.NTOTEL) THEN
           WRITE(105,*) ETTOT + ESPR
           ITIME = 0
           REWIND(2000)
           REWIND(6000)
        ENDIF

        RETURN
        END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  P R O G R A M                                                     .
C .       TO CALCULATE                                                 .
C .            SPRING STIFFNESS MATRIX                                 .
C .            SPRING RESIDUAL FORCE VECTOR                            .
C .       OF SPRNIG ELEMNTS (TYPE III IV, AND V)                       .
C .                                                                    .
C .  N O T E                                                           .
C .        INPUT  : SPRINGEX, SPRINGEX0, NEE, JTYPE                    .
C .        OUTPUT : SPRINGEK, SPRINGEF, ESPR                           .
```

```
C .                                                                   .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      SUBROUTINE SPRINGEMEF(SPRINGEX,SPRINGEX0,SPRINGEK,SPRINGEF,ESPR,
     &                      NEE)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 SPRINGEX(2,3),SPRINGEX0(2,3),SPRINGEK(6,6),SPRINGEF(6)

      STIFF = 1.0D10

      X1 = SPRINGEX(1,1)
      Y1 = SPRINGEX(1,2)
      Z1 = SPRINGEX(1,3)
      X2 = SPRINGEX(2,1)
      Y2 = SPRINGEX(2,2)
      Z2 = SPRINGEX(2,3)

      R = DSQRT((X2-X1)**2.D0 + (Y2-Y1)**2.D0 + (Z2-Z1)**2.D0)
      R0 = DSQRT((SPRINGEX0(2,1)-SPRINGEX0(1,1))**2.D0 +
(SPRINGEX0(2,2)
     &      -SPRINGEX0(1,2))**2.D0 + (SPRINGEX0(2,3)-
SPRINGEX0(1,3))**2.D0)
C
C  FOR DEBUG PURPOSE
C      IF(DABS(R-R0).GE.1D-3) THEN
C            WRITE(7,743) NEE, R, R0, R - R0
C743         FORMAT(i5,',',3f20.15)
C     ENDIF
C
      ESPR = ESPR + 0.5D0*STIFF*(R-R0)**2.D0
      DR1 = STIFF*(1.D0-R0/R)
      DDR2 = STIFF*R0/R**3.D0
C
C  CONSTRUCTING THE SPRING STIFFNESS MATRIX AND RESIDUAL FORCE VECTOR
C
      SPRINGEF(1) = DR1*(X1-X2)
      SPRINGEF(2) = DR1*(Y1-Y2)
      SPRINGEF(3) = DR1*(Z1-Z2)
      SPRINGEF(4) = -DR1*(X1-X2)
      SPRINGEF(5) = -DR1*(Y1-Y2)
      SPRINGEF(6) = -DR1*(Z1-Z2)

      SPRINGEK(1,1) = DDR2*(X1-X2)*(X1-X2)+DR1
      SPRINGEK(1,2) = DDR2*(X1-X2)*(Y1-Y2)
      SPRINGEK(1,3) = DDR2*(X1-X2)*(Z1-Z2)
      SPRINGEK(1,4) = -SPRINGEK(1,1)
      SPRINGEK(1,5) = -SPRINGEK(1,2)
      SPRINGEK(1,6) = -SPRINGEK(1,3)

      SPRINGEK(2,1) = SPRINGEK(1,2)
      SPRINGEK(2,2) = DDR2*(Y1-Y2)*(Y1-Y2)+DR1
      SPRINGEK(2,3) = DDR2*(Y1-Y2)*(Z1-Z2)
      SPRINGEK(2,4) = -SPRINGEK(2,1)
      SPRINGEK(2,5) = -SPRINGEK(2,2)
      SPRINGEK(2,6) = -SPRINGEK(2,3)

      SPRINGEK(3,1) = SPRINGEK(1,3)
      SPRINGEK(3,2) = SPRINGEK(2,3)
```

```
      SPRINGEK(3,3) = DDR2*(Z1-Z2)*(Z1-Z2)+DR1
      SPRINGEK(3,4) = -SPRINGEK(3,1)
      SPRINGEK(3,5) = -SPRINGEK(3,2)
      SPRINGEK(3,6) = -SPRINGEK(3,3)

      SPRINGEK(4,1) = SPRINGEK(1,4)
      SPRINGEK(4,2) = SPRINGEK(2,4)
      SPRINGEK(4,3) = SPRINGEK(3,4)
      SPRINGEK(4,4) = -SPRINGEK(4,1)
      SPRINGEK(4,5) = -SPRINGEK(4,2)
      SPRINGEK(4,6) = -SPRINGEK(4,3)

      SPRINGEK(5,1) = SPRINGEK(1,5)
      SPRINGEK(5,2) = SPRINGEK(2,5)
      SPRINGEK(5,3) = SPRINGEK(3,5)
      SPRINGEK(5,4) = -SPRINGEK(5,1)
      SPRINGEK(5,5) = -SPRINGEK(5,2)
      SPRINGEK(5,6) = -SPRINGEK(5,3)

      SPRINGEK(6,1) = SPRINGEK(1,6)
      SPRINGEK(6,2) = SPRINGEK(2,6)
      SPRINGEK(6,3) = SPRINGEK(3,6)
      SPRINGEK(6,4) = -SPRINGEK(6,1)
      SPRINGEK(6,5) = -SPRINGEK(6,2)
      SPRINGEK(6,6) = -SPRINGEK(6,3)
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  P R O G R A M                                                     .
C .       TO CALCULATE                                                 .
C .            ELEMENTAL STIFFNESS MATRIX                              .
C .            ELEMENTAL RESIDUAL VECTOR                               .
C .       OF TYPE I AND II ELEMENTS                                    .
C .                                                                    .
C .  N O T E                                                           .
C .        INPUT  : EX, NELE, JELEM                                    .
C .        OUTPUT : EK, EF, ETOT1                                      .
C .        STIFFNESS MATRIX, EK :                                      .
C .              IT IS SYMMETRIC                                       .
C .              IT OBTAINED USING,                                    .
C .              0.5 Partial V / (Partial (X(1)_L) Partial (X(I)_J))   .
C .              THE COEFFICENT 0.5 IS USED TO AVOID DOUBLE COUNT OF   .
C .              OF ENERGY FROM THE SAME TETRA HEDERAL ANGLE.          .
C .              ITS SIZE IS 27X27                                     .
C .        UNEQUILIBRIUM FORCE VECTOR, EF :                            .
C .              IT OBTAINED USING,                                    .
C .              Partial V / (Partial (X(1)_L) AT THE CENTRE VERTEX    .
C .              ITS SIZE IS 27X1                                      .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      SUBROUTINE CALEMEF(EX,NELE,EK,EF,ETTOT,JELEM)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 EX(9,3),EK(27,27),EF(27)
      REAL*8 X(3,3),PVPX(3),PPVPXPX(3,3,3)
      INTEGER NELE(9),NN(3)
      PARAMETER (STIFF1=100.D0,STIFF2=100.D0)
```

```
C
C  EQUILIBRUM VALUES OF PHI AND BETA
C
      PI = DACOS(-1.D0)
      PHI1 = PI/180.D0*(60.D0)
      BETA = PI/180.D0*(45.D0)

      ALPHA1 = DACOS(1 - (2.D0*DSIN(PHI1/2.D0)**2.D0)/
     &                (DSIN(BETA)**2.D0))
      ALPHA2 = DACOS(1 - (2.D0*DTAN(PHI1/2.D0)**2.D0)/
     &                (DTAN(BETA)**2.D0))

      EK = 0.D0
      EF = 0.D0
      ET1 = 0.D0
C
C  CHECKING FOR EXISTANCE OF ANGLE AND ASSIGNING CORRESPONDING
C  PARAMETERS TO STIFF AND EANGLE FOR CONSTRUCTING THE ELEMENTAL
C  STIFFNESS MATRIX AND RESIDUAL FORCE
C
      DO IANGLE = 1,6
         NN = 1
         IF(IANGLE.EQ.1.AND.NELE(2).NE.1.AND.NELE(3).NE.1) THEN
            NN(1) = 2
            NN(2) = 1
            NN(3) = 3
            STIFF = STIFF2
            EANGLE = ALPHA2
         ENDIF
         IF(IANGLE.EQ.2.AND.NELE(4).NE.1.AND.NELE(5).NE.1) THEN
            NN(1) = 4
            NN(2) = 1
            NN(3) = 5
            STIFF = STIFF1
            EANGLE = ALPHA1
         ENDIF
         IF(IANGLE.EQ.3.AND.NELE(2).NE.1.AND.NELE(6).NE.1) THEN
            NN(1) = 1
            NN(2) = 2
            NN(3) = 6
            STIFF = STIFF2
            EANGLE = ALPHA2
         ENDIF
         IF(IANGLE.EQ.4.AND.NELE(3).NE.1.AND.NELE(7).NE.1) THEN
            NN(1) = 1
            NN(2) = 3
            NN(3) = 7
            STIFF = STIFF2
            EANGLE = ALPHA2
         ENDIF
         IF(IANGLE.EQ.5.AND.NELE(4).NE.1.AND.NELE(8).NE.1) THEN
            NN(1) = 1
            NN(2) = 4
            NN(3) = 8
            STIFF = STIFF1
            EANGLE = ALPHA1
         ENDIF
```

```
          IF(IANGLE.EQ.6.AND.NELE(5).NE.1.AND.NELE(9).NE.1) THEN
             NN(1) = 1
             NN(2) = 5
             NN(3) = 9
             STIFF = STIFF1
             EANGLE = ALPHA1
          ENDIF

          IF(NN(1) + NN(2) + NN(3).NE.3) THEN
             DO II = 1,3
                X(II,:) = EX(NN(II),:)
             ENDDO
C
C  CALCULATING THE PARTIAL DERIVATES OF ENERGY (V)
C
             CALL
DIFF(IANGLE,STIFF,EANGLE,X,NN,BETA,PVPX,PPVPXPX,ET,JELEM)

             ETTOT = ETTOT + ET
             DO L = 1,3
                EF(L) = EF(L) + PVPX(L)
             ENDDO
             DO L = 1,3
                DO I = 1,3
                   DO J = 1,3
                      IF(NN(I).EQ.1) THEN
                         EK(L,J) = EK(L,J) + PPVPXPX(L,I,J)
                      ELSE
                         EK(L,3*(NN(I)-1)+J) = EK(L,3*(NN(I)-1)+J)
     &                                  +0.5*PPVPXPX(L,I,J)
                         EK(3*(NN(I)-1)+J,L) = EK(L,3*(NN(I)-1)+J)
                      ENDIF
                   ENDDO
                ENDDO
             ENDDO
          ENDIF
       ENDDO
       RETURN
       END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  P R O G R A M                                                     .
C .       TO CALCULATE FIRST TWO DERIVATIVE OF STAIN ENERGY            .
C .                                                                    .
C .  N O T E                                                           .
C .        INPUT  : IANGLE, STIFF, EANGLE, X, NSE, JTYPE, JELEM       .
C .        OUTPUT : PVPX, PPVPXPX, ET                                  .
C .        REFER THESIS FOR THE FORMULAS USED IN THIS SECTION          .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
       SUBROUTINE DIFF(IANGLE,STIFF,EANGLE,X,NSE,BETA,PVPX,PPVPXPX,ET,
      1                JELEM)
       IMPLICIT DOUBLE PRECISION (A-H, O-Z)
       REAL*8 X(3,3),R(3,3),PCPX(3,3),PVPX(3),PRPX(3,3,3,3)
       REAL*8 PPCPXPX(3,3,3,3),PPVPXPX(3,3,3)
       INTEGER NSE(3)
```

```
      DO I = 1,3
         DO J = 1,3
            R(I,J) = DISTANCE(X(I,:),X(J,:))
         ENDDO
      ENDDO

      ET = 0.D0
      PI = DACOS(-1.D0)

      COS_THETA = CFC(R(1,2),R(2,3),R(1,3))
      SIN_THETA = DSQRT(1.D0 - COS_THETA**2.D0)
      THETA = DACOS(COS_THETA)

      IF(IANGLE.EQ.2 .OR. IANGLE.EQ.5 .OR. IANGLE.EQ.6) THEN
C
C  THE FOLLWING STEP IS LOGICAL LIMIT, TO PREVENT THE PROGRAM FROM
C  CRASHING. NUMERICAL ROUND OFF SOMETIMES MAKES COS AND SIN > 1
C
         IF(THETA.GE.PI/2.D0) THETA = PI/2.D0-0.0001D0
         ALPHA = DACOS(1.D0 - (2.D0*DSIN(THETA/2.D0)**2)/
     &             (DSIN(BETA)**2))

         PAPT = (DSQRT(2.D0)*DCOS(THETA/2.D0))/
     &            DSQRT(DCOS(THETA) - DCOS(2*BETA))
         PPAPTPT = (DSQRT(2.D0)*DCOS(BETA)**2.D0)/
     &           ((DCOS(THETA) - DCOS(2*BETA))**1.5D0*
     &         (DSIN(THETA/2))**2)

         IF(IANGLE.EQ.2) ET = 0.5D0*STIFF*(ALPHA-EANGLE)**2.D0
      ENDIF

      IF(IANGLE.EQ.1 .OR. IANGLE.EQ.3 .OR. IANGLE.EQ.4) THEN
C
C  LOGICAL LIMIT
C
         IF(THETA.GE.PI) THETA = PI-0.0001D0
         ALPHA = DACOS((DCOS(THETA) + DCOS(BETA)**2)/
     &             (DSIN(BETA)**2))

         PAPT = (DSIN(THETA))/(DSQRT(DSIN(BETA)**4
     &           -(DCOS(BETA)**2 + DCOS(THETA))**2))
         PPAPTPT = (2.D0*DCOS(THETA/2)**2*DCOS(BETA)**2)/
     &           ((DCOS(2*BETA) + DCOS(THETA))*DSQRT(DSIN(BETA)**4
     &        -(DCOS(BETA)**2 + DCOS(THETA))**2))

         IF(IANGLE.EQ.1) ET = 0.5D0*STIFF*(ALPHA-EANGLE)**2.D0
      ENDIF

      PVPA = STIFF*(ALPHA-EANGLE)
      PPVPAPA = STIFF

      PTPC = -1.D0/SIN_THETA
      PPTPCPC = -COS_THETA/SIN_THETA**3.D0

      DO I = 1,3
         DO J = 1,3
```

```
            PCPX(I,J) =
     &        PCPR1FC(R(1,2),R(2,3),R(1,3))
     &      * PRPXFC(1,2,I,J,X(1,:),X(2,:),R(1,2))
     &      + PCPR2FC(R(1,2),R(2,3),R(1,3))
     &      * PRPXFC(2,3,I,J,X(2,:),X(3,:),R(2,3))
     &      + PCPR3FC(R(1,2),R(2,3),R(1,3))
     &      * PRPXFC(1,3,I,J,X(1,:),X(3,:),R(1,3))
         ENDDO
       ENDDO
C
C  FOR DEBUG PURPOSES
C        IF (JELEM.eq.70) THEN
C               WRITE(7,765) iangle,theta,alpha,et
C               WRITE(7,765) iangle,pvpa,papt,ptpc
C765            FORMAT(i8,3f26.20)
C        ENDIF
C
      IF(NSE(1).EQ.1) PVPX = PVPA*PAPT*PTPC*PCPX(1,:)
      IF(NSE(2).EQ.1) PVPX = PVPA*PAPT*PTPC*PCPX(2,:)

      PCPR12 = PCPR1FC(R(1,2),R(2,3),R(1,3))
      PCPR23 = PCPR2FC(R(1,2),R(2,3),R(1,3))
      PCPR13 = PCPR3FC(R(1,2),R(2,3),R(1,3))
      PPCPR12PR12 = PPCPRPR11FC(R(1,2),R(2,3),R(1,3))
      PPCPR12PR23 = PPCPRPR12FC(R(1,2),R(2,3),R(1,3))
      PPCPR12PR13 = PPCPRPR13FC(R(1,2),R(2,3),R(1,3))
      PPCPR23PR23 = PPCPRPR22FC(R(1,2),R(2,3),R(1,3))
      PPCPR23PR13 = PPCPRPR23FC(R(1,2),R(2,3),R(1,3))
      PPCPR13PR13 = PPCPRPR33FC(R(1,2),R(2,3),R(1,3))

      PRPX = 0.
      DO I = 1,3
         DO J = 1,3
            DO K = 1,3
               DO L = 1,3
                  PRPX(I,J,K,L) = PRPXFC(I,J,K,L,X(I,:),X(J,:),R(I,J))
               ENDDO
            ENDDO
         ENDDO
      ENDDO

      DO I = 1,3
         DO J = 1,3
            DO K = 1,3
               DO L = 1,3
           PPCPXPX(I,J,K,L) = (PPCPR12PR12*PRPX(1,2,K,L) + PPCPR12PR23
     &             *PRPX(2,3,K,L) +
PPCPR12PR13*PRPX(1,3,K,L))*PRPX(1,2,I,J)
     &        + PCPR12*PPRPXPXFC(1,2,I,K,J,L,X(1,:),X(2,:),R(1,2))
     &        + (PPCPR12PR23*PRPX(1,2,K,L)+PPCPR23PR23
     &            *PRPX(2,3,K,L) +
PPCPR23PR13*PRPX(1,3,K,L))*PRPX(2,3,I,J)
     &        + PCPR23*PPRPXPXFC(2,3,I,K,J,L,X(2,:),X(3,:),R(2,3))
     &        + (PPCPR12PR13*PRPX(1,2,K,L)+PPCPR23PR13
     &            *PRPX(2,3,K,L) +
PPCPR13PR13*PRPX(1,3,K,L))*PRPX(1,3,I,J)
     &        + PCPR13*PPRPXPXFC(1,3,I,K,J,L,X(1,:),X(3,:),R(1,3))
```

```
                ENDDO
              ENDDO
            ENDDO
          ENDDO

          IF(NSE(1).EQ.1) M = 1
          IF(NSE(2).EQ.1) M = 2
          DO J = 1,3
             DO K = 1,3
                DO L = 1,3
            PPVPXPX(J,K,L) = PPVPAPA*PAPT**2.D0*PTPC**2.D0*PCPX(M,J)*
     &      PCPX(K,L) + PVPA*PPAPTPT*PTPC**2.D0*PCPX(M,J)*PCPX(K,L)
     &        + PVPA*PAPT*PPTPCPC*PCPX(M,J)*PCPX(K,L)
     &          + PVPA*PAPT*PTPC*PPCPXPX(M,J,K,L)
                ENDDO
             ENDDO
          ENDDO

          RETURN
          END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  F U N C T I O N                                                   .
C .       TO CALCULATE DISTANE BETWEEN TWO POINTS                      .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
          FUNCTION DISTANCE(X1,X2)
          IMPLICIT DOUBLE PRECISION (A-H, O-Z)
          REAL*8 DISTANCE
          REAL*8 X1(3),X2(3)
          DISTANCE = DSQRT((X1(1)-X2(1))**2.D0 + (X1(2)-X2(2))**2.D0
     &            + (X1(3)-X2(3))**2.D0)
          RETURN
          END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  F U N C T I O N                                                   .
C .       TO CALCULATE COS ANGLE FORMED BY B, AND C                    .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
          FUNCTION CFC(B,C,A)
          IMPLICIT DOUBLE PRECISION (A-H, O-Z)
          REAL*8 CFC
          CFC = (B*B + C*C - A*A)/(2.D0*B*C)
          RETURN
          END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  F U N C T I O N                                                   .
C .       TO CALCULATE FIRST DERIVATIVE OF COS FUNCTION                .
C .                                                                    .
C .  N O T E                                                           .
C .        COS(ANGLE) = (B^2+C^2-A^2/(2*B*C)                           .
C .        THE DERIVATIVE IS WITH REPECT TO B                          .
```

```
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PCPR1FC(B,C,A)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PCPR1FC
      PCPR1FC = 0.5*(B**2.D0 - C**2.D0 + A**2.D0)/B**2.D0/C
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  F U N C T I O N                                                   .
C .       TO CALCULATE FIRST DERIVATIVE OF COS FUNCTION                .
C .                                                                    .
C .  N O T E                                                           .
C .        COS(ANGLE) = (B^2+C^2-A^2/(2*B*C)                           .
C .        THE DERIVATIVE IS WITH REPECT TO C                          .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PCPR2FC(B,C,A)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PCPR2FC
      PCPR2FC = 0.5*(C**2.D0 - B**2.D0 + A**2.D0)/B/C**2.D0
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  F U N C T I O N                                                   .
C .       TO CALCULATE FIRST DERIVATIVE OF COS FUNCTION                .
C .                                                                    .
C .  N O T E                                                           .
C .        COS(ANGLE) = (B^2+C^2-A^2/(2*B*C)                           .
C .        THE DERIVATIVE IS WITH REPECT TO A                          .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PCPR3FC(B,C,A)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PCPR3FC
      PCPR3FC = -A/B/C
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  F U N C T I O N                                                   .
C .       TO CALCULATE SECOND DERIVATIVE OF COS FUNCTION               .
C .                                                                    .
C .  N O T E                                                           .
C .        COS(ANGLE) = (B^2+C^2-A^2/(2*B*C)                           .
C .        THE DERIVATIVE IS WITH REPECT TO BB                         .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PPCPRPR11FC(B,C,A)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PPCPRPR11FC
      PPCPRPR11FC = (C**2.D0 - A**2.D0)/(B**3.D0*C)
```

```
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  F U N C T I O N                                                   .
C .       TO CALCULATE SECOND DERIVATIVE OF COS FUNCTION               .
C .                                                                    .
C .  N O T E                                                           .
C .        COS(ANGLE) = (B^2+C^2-A^2/(2*B*C)                           .
C .        THE DERIVATIVE IS WITH REPECT TO BC                         .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PPCPRPR12FC(B,C,A)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PPCPRPR12FC
      PPCPRPR12FC = -(B**2.D0+C**2.D0+A**2.D0)/(2.D0*B**2.D0*C**2.D0)
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  F U N C T I O N                                                   .
C .       TO CALCULATE SECOND DERIVATIVE OF COS FUNCTION               .
C .                                                                    .
C .  N O T E                                                           .
C .        COS(ANGLE) = (B^2+C^2-A^2/(2*B*C)                           .
C .        THE DERIVATIVE IS WITH REPECT TO BA                         .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PPCPRPR13FC(B,C,A)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PPCPRPR13FC
      PPCPRPR13FC = A/(B**2.D0*C)
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  F U N C T I O N                                                   .
C .       TO CALCULATE SECOND DERIVATIVE OF COS FUNCTION               .
C .                                                                    .
C .  N O T E                                                           .
C .        COS(ANGLE) = (B^2+C^2-A^2/(2*B*C)                           .
C .        THE DERIVATIVE IS WITH REPECT TO CC                         .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PPCPRPR22FC(B,C,A)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PPCPRPR22FC
      PPCPRPR22FC = (B**2.D0 - A**2.D0)/(B*C**3.D0)
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  F U N C T I O N                                                   .
```

```
C .       TO CALCULATE SECOND DERIVATIVE OF COS FUNCTION              .
C .                                                                   .
C .  N O T E                                                          .
C .       COS(ANGLE) = (B^2+C^2-A^2/(2*B*C)                           .
C .       THE DERIVATIVE IS WITH REPECT TO CA                         .
C .                                                                   .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PPCPRPR23FC(B,C,A)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PPCPRPR23FC
      PPCPRPR23FC = A/(B*C**2.D0)
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                   .
C .  F U N C T I O N                                                  .
C .      TO CALCULATE SECOND DERIVATIVE OF COS FUNCTION               .
C .                                                                   .
C .  N O T E                                                          .
C .       COS(ANGLE) = (B^2+C^2-A^2/(2*B*C)                           .
C .       THE DERIVATIVE IS WITH REPECT TO AA                         .
C .                                                                   .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PPCPRPR33FC(B,C,A)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PPCPRPR33FC
      PPCPRPR33FC = -1.D0/(B*C)
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                   .
C .  F U N C T I O N                                                  .
C .      TO CALCULATE Partial R / Partial X                           .
C .                                                                   .
C .  N O T E                                                          .
C .       R : The distance of atom I and J                            .
C .       K : If K=I, the derivative is respect to atom I             .
C .         : If K=J, the derivative is respect to atom J             .
C .       L : Direction of the derivative. 1, 2, and 3 means          .
C .           X, Y, and Z                                             .
C .                                                                   .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PRPXFC(I,J,K,L,XI,XJ,R)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PRPXFC,XI(3),XJ(3)
      IF((I.EQ.J).OR.((K.NE.I).AND.(K.NE.J))) THEN
         PRPXFC = 0.
         RETURN
      ENDIF
      IF(K.EQ.I) THEN
         PRPXFC = (XI(L) - XJ(L))/R
         RETURN
      ENDIF
      IF(K.EQ.J) THEN
         PRPXFC = (XJ(L) - XI(L))/R
```

```
         RETURN
      ENDIF
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  F U N C T I O N                                                   .
C .       TO CALCULATE                                                 .
C .       Partial (R_IJ) / Partial (X(K)_M) Partial (X(L)_N)           .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PPRPXPXFC(I,J,K,L,M,N,XK,XL,RIJ)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PPRPXPXFC,XK(3),XL(3)
      IF((K.NE.I.AND.K.NE.J).OR.(L.NE.I.AND.L.NE.J)) THEN
         PPRPXPXFC = 0.D0
         RETURN
      ENDIF
      IF(K.EQ.L) THEN
         IF(M.EQ.N) THEN
            PPRPXPXFC = 1.D0/RIJ - (XK(M) - XL(M))**2.D0/RIJ**3.D0
            RETURN
         ENDIF
         IF(M.NE.N) THEN
            PPRPXPXFC = -(XK(M) - XL(M))*(XK(N) - XL(N))/RIJ**3.D0
            RETURN
         ENDIF
      ENDIF
      IF(K.NE.L) THEN
         IF(M.EQ.N) THEN
            PPRPXPXFC = -1.D0/RIJ + (XK(M) - XL(M))**2.D0/RIJ**3.D0
            RETURN
         ENDIF
         IF(M.NE.N) THEN
            PPRPXPXFC = (XK(M) - XL(M))*(XK(N) - XL(N))/RIJ**3.D0
            RETURN
         ENDIF
      ENDIF
      PPRPXPXFC = 0.
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C . P A R A M E T E R S   U S E D   A N D   T H I E R   M E A N I N G .
C .                                                                    .
C . AE         : ARRAY CONTAINING THE EQUILIBRIUM VALUES OF THE 9      .
C .              DIHEDERAL                                             .
C .              ANGLES IN THE ELEMENT                                 .
C . AMATRX     : JACOBIAN MATRIX, DEPENDING ON THE SIMULATION IT CAN   .
C .              BE K OR M                                             .
C .              MATRIX                                                .
C . COORDS     : ARRAY CONTAINING THE ORIGINAL COORDINATES OF THE      .
C .              ELEMENT                                               .
C . DR1        : FIRST DERIVATIVE WRT R                                .
C . DDR2       : SECOND DERIVATIVE WRT R                               .
```

```
C . EF        : ELEMENTAL UNBALCAED FORCE VECTOR                        .
C . EK        : ELEMENTAL STIFFNESS MATRIX                              .
C . ESPR      : TOTAL ENERGY OF SPRING ELEMENTS                         .
C . ETTOT     : TOTAL ENERGY IN THE 19 NODE ELEMENTS                    .
C . EX        : ARRAY CONTAINING THE CURRENT COORDINATES OF THE         .
C .             ELEMENT                                                 .
C . IMCRD     : MAXIMUM NUMBER OF COORDINATES NEEDED TO REPRESENT       .
C .             A NODE (VALUE IS 3)                                     .
C . ITIME     : COUNTER THAT KEEPS TRACK OF NUMBER OF ELEMENTS          .
C .             SIMULATED IN AN ITERATION                               .
C . JTYPE     : THE TYPE OF ELEMENT IN THE CURRENT LOOP                 .
C . JELEM     : ELEMENT NUMBER ASSIGNED TO THE CURRENT ELEMENT          .
C . LFLAGS    : LFLAGS(1) = 1  -> STATIC ANALYSIS                       .
C .             LFLAGS(1) = 41 -> EIGEN FREQUENCY ANALYSIS              .
C .             LFLAGS(1) = 2  -> STATIC ANALYSIS                       .
C .             LFLAGS(3) = 2  -> DEFINES CURRENT STIFFNESS MATRIX      .
C .             LFLAGS(3) = 4  -> DEFINES CURRENT MASS MATRIX           .
C . MASS      : MASS MATRIX                                             .
C . NDOFEL    : NUMBER OF DEGREES OF FREEDOM FOR THE ELEMENT            .
C .             (VALUE = 27)                                            .
C . NELE      : ARRAY STORING THE GLOBAL ELEMENT NODE NUMBERS           .
C . NN        : ARRAY CONTAINING THE NODE NUMBERS OF THE 3 NODES        .
C .             CONTRIBUTING AN ANGLE UNDER CONSIDERATION               .
C . NNODE     : NUMBER OF NODES IN THE ELEMENT                          .
C .             (VALUE = 9)                                             .
C . NTOTEL    : TOTAL NUMBER OF ELEMENTS                                .
C . PVPA      : PARTIAL DEIVATIVE OF ENERGY WRT TO A. USING THE SAME    .
C .             CONVETION THE NOTATIONS SUCH AS, PPVPAPA, PCPX, ETC.    .
C .             CAN BE INFERRED                                         .
C . PPVPXPX   : SECOND DERIVATIV.E OF ENERGY(V) WRT POSITOIN OF THE     .
C .             CENTRAL ATOM(X)                                         .
C . PVPX      : FIRST DERIVATIVE OF ENERGY(V) WRT POSITOIN OF THE       .
C .             CENTRAL ATOM(X)                                         .
C . R         : LENGTH OF BOND BEFORE DEFORMATION                       .
C . RHS       : RIGHT HAND SIDE RESIDUAL VECTOR                         .
C . R0        : ORIGINAL LENGTH OF BOND                                 .
C . R0REF     : ARRAY STROING THE BOND LENTHS OF ELEMENTS III, IV,      .
C .             AND V                                                   .
C . SHIFT     : POTENTIAL ENERGY SHIFT TO ENSURE THE TOTAL ENERGY       .
C .             IS ZERO                                                 .
C . SPRINGRF  : SPRING UNABALNCE FORCE VECTOR                           .
C . SPRINGEK  : SPRING STIFFNESS MATRIX                                 .
C . SPRINGEX  : ARRAY CONTAINING THE CURRENT COORDINATES OF THE         .
C .             SPRING ELEMENT                                          .
C . SPRINGEX0 : ARRAY CONTAINING THE ORIGINAL COORDINATES OF THE        .
C .             SPRING ELEMENT                                          .
C . STIFF     : STIFFNESS OF THE CREASE OR BOND                         .
C . U         : DISPLACEMENT INCREMENT                                  .
C . X         : ARRAY CONTAINING THE COORDINATES OF THE 3 NODES         .
C .             CONTRIBUTING AN ANGLE UNDER CONSIDERATION               .
C .                                                                     .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
```

UEL FOR RR ELEMENT TYPE

```
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .       U E L  F O R  N O N  P E R I O D I C  O R I G A M I          .
C .              R O N  R E S C H  P A T T E R N                       .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

      SUBROUTINE UEL(RHS,AMATRX,SVARS,ENERGY,NDOFEL,NRHS,NSVARS,
     1 PROPS,NPROPS,COORDS,MCRD,NNODE,U,DU,V,A,JTYPE,TIME,DTIME,
     2 KSTEP,KINC,JELEM,PARAMS,NDLOAD,JDLTYP,ADLMAG,PREDEF,NPREDF,
     3 LFLAGS,MLVARX,DDLMAG,MDLOAD,PNEWDT,JPROPS,NJPROP,PERIOD)

      INCLUDE 'ABA_PARAM.INC'
      REAL*8 KB
      PARAMETER (NTOTEL = 1089)
      PARAMETER ( SHIFT = 0.0 )

      DIMENSION RHS(MLVARX,*),AMATRX(NDOFEL,NDOFEL),PROPS(*),
     1 SVARS(NSVARS),ENERGY(8),COORDS(MCRD,NNODE),U(NDOFEL),
     2 DU(MLVARX,*),V(NDOFEL),A(NDOFEL),TIME(2),PARAMS(*),
     3 JDLTYP(MDLOAD,*),ADLMAG(MDLOAD,*),DDLMAG(MDLOAD,*),
     4 PREDEF(2,NPREDF,NNODE),LFLAGS(*),JPROPS(*)

      REAL*8 EX(19,3),EK(57,57),EF(57)
      REAL*8 SPRINGEX(2,3),SPRINGEX0(2,3),SPRINGEK(6,6),SPRINGEF(6)
      INTEGER NELE(19)
      COMMON ITIME
      COMMON ETTOT
      REAL*8 MASS

      MASS = 1.D0
      OPEN(2000,FILE = 'C:\Users\Deepak\Desktop\cheng\coors_RR.TXT')
      OPEN(105,FILE = 'C:\Users\Deepak\Desktop\cheng\energy_RR.dat')
      OPEN(6000,FILE = 'C:\Users\Deepak\Desktop\cheng\warning_RR.dat')

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C . E L E M E N T  A R R A Y  S E Q U E N C E                          .
C .                                                                    .
C .                                                                    .
C . NELE(1) = 1 (always)                                               .
C . NELE(2) = 2                                                        .
C . NELE(3) = 3                                                        .
C . NELE(4) = 4                                                        .
C . NELE(5) = 5                                                        .
C . NELE(6) = 6                                                        .
C . NELE(7) = 7                                                        .
C . NELE(8) = 8                                                        .
C . NELE(9) = 9                                                        .
C . NELE(10) = 10                                                      .
C . NELE(11) = 11                                                      .
C . NELE(12) = 12                                                      .
C . NELE(13) = 13                                                      .
C . NELE(14) = 14                                                      .
C . NELE(15) = 15                                                      .
```

```
C . NELE(16) = 16                                                      .
C . NELE(17) = 17                                                      .
C . NELE(18) = 18                                                      .
C . NELE(19) = 19                                                      .
C . IF NELE(I) = 1, FOR I = 2 TO 19, IT MEANS THAT THE NODE DOES NOT   .
C . EXIST                                                              .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

C
C  RESET THE TOTAL ENERGY IN ALL THE ELEMENTS TO ZERO AFTER EVERY
C  ITERATION
C
      IF(ITIME.EQ.0) THEN
         ETTOT = 0.D0
         ESPR = 0.D0
      ENDIF

      AMATRX = 0.D0

      DO K1 = 1,NDOFEL
             DO KRHS = 1,NRHS
                  RHS(K1,KRHS) = 0.D0
             ENDDO
      ENDDO

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C . E L E M E N T  T Y P E S                                           .
C .                                                                    .
C . JTYPE = 1: 19 NODES, 6 DIHEDERAL ANGLES                            .
C . JTYPE = 2: 19 NODES, 9 DIHEDERAL ANGLES                            .
C . JTYPE = 3:  2 NODES, LENGTH = 1                                    .
C . JTYPE = 4:  2 NODES, LENGTH = 2/SQRT(3)                            .
C . JTYPE = 5:  2 NODES, LENGTH = 1/SQRT(3)                            .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

C
C  CALCULATING THE CURRENT COORDINATES
C
      IF(JTYPE.EQ.1.OR.JTYPE.EQ.2) THEN
             DO INODE = 1,NNODE
                  DO IMCRD = 1,MCRD
                   IPOSN = IMCRD + 3*(INODE-1)
                     EX(INODE,IMCRD) = COORDS(IMCRD,INODE) + U(IPOSN)
                ENDDO
             ENDDO
             NELE = 1
C
C  CHECK IF NODE EXISTS, I.E., IF ITS DISTANCE FROM CENTRAL NODE IS
C  GREATER THAN 0
C
             DO INODE = 2,NNODE
                DIS = DSQRT((EX(INODE,1)-EX(1,1))**2.D0
     &       + (EX(INODE,2)-EX(1,2))**2.D0 + (EX(INODE,3)-
EX(1,3))**2.D0)
```

```
                  IF(DIS.LT.1.D-7) THEN
                       NELE(INODE) = 1
                  ELSE
                       NELE(INODE) = INODE
                  ENDIF
               ENDDO
C
C  CALLING CALEMEF TO OBTAIN ELEMENTAL STIFFNESS AND FORCE VECTOR
C
               ETTOT1 = ETTOT
               CALL CALEMEF(EX,NELE,EK,EF,ETTOT1,JTYPE)
               ETTOT = ETTOT1
       ENDIF

       IF(JTYPE.EQ.3.OR.JTYPE.EQ.4.OR.JTYPE.EQ.5) THEN
               DO INODE = 1,NNODE
                    DO IMCRD = 1,MCRD
                         IPOSN = IMCRD + 3*(INODE-1)
                         SPRINGEX(INODE,IMCRD) = COORDS(IMCRD,INODE) +
U(IPOSN)
                         SPRINGEX0(INODE,IMCRD) = COORDS(IMCRD,INODE)
                    ENDDO
               ENDDO
C
C  CALLING SPRINGEMEF TO OBTAIN SPRING STIFFNESS AND FORCE VECTOR
C
               CALL SPRINGEMEF(SPRINGEX,SPRINGEX0,SPRINGEK,SPRINGEF,ESPR,
     &                         JELEM,JTYPE)
       ENDIF

       IF(LFLAGS(1).EQ.1 .OR. LFLAGS(1).EQ.2) THEN
               IF(JTYPE.EQ.1.OR.JTYPE.EQ.2) THEN
                    RHS(1,1) = -EF(1)
                    RHS(2,1) = -EF(2)
                    RHS(3,1) = -EF(3)
                    DO K1 = 1,NDOFEL
                       DO K2 = 1,NDOFEL
                            AMATRX(K2,K1) = EK(K2,K1)
                       ENDDO
                    ENDDO
C
C  A CONSTANT IS ADDED TO THE DIAGONAL TERMS TO MAKE THE AMATRIX
C  POSITIVE DEFINTIE
C
                    AMATRX(1,1) = AMATRX(1,1) + 0D6
                    AMATRX(2,2) = AMATRX(2,2) + 0D6
                    AMATRX(3,3) = AMATRX(3,3) + 0D6
                    WRITE(2000,2000) JELEM,EX(1,1),EX(1,2),EX(1,3)
2000                FORMAT(I8,',',f20.10,',',f20.10,',',f20.10)
               ENDIF
               IF(JTYPE.EQ.3.OR.JTYPE.EQ.4.OR.JTYPE.EQ.5) THEN
                    RHS(1,1) = -SPRINGEF(1)
                    RHS(2,1) = -SPRINGEF(2)
                    RHS(3,1) = -SPRINGEF(3)
                    RHS(4,1) = -SPRINGEF(4)
                    RHS(5,1) = -SPRINGEF(5)
                    RHS(6,1) = -SPRINGEF(6)
```

```
                  DO K1 = 1,NDOFEL
                     DO K2 = 1,NDOFEL
                          AMATRX(K2,K1) = SPRINGEK(K2,K1)
                     ENDDO
                  ENDDO
            ENDIF
      ENDIF
C
C  THE VALUE OF LFLAGS DETERMINES THE TYPE OF SIMULATION
C
      IF(LFLAGS(1).EQ.41) THEN
            IF(LFLAGS(3).EQ.2) THEN
                  IF(JTYPE.EQ.1.OR.JTYPE.EQ.2) THEN
                     DO K1 = 1,NDOFEL
                          DO K2 = 1,NDOFEL
                               AMATRX(K2,K1) = EK(K2,K1)
                          ENDDO
                        ENDDO
                  ENDIF
                  IF(JTYPE.EQ.3.OR.JTYPE.EQ.4.OR.JTYPE.EQ.5) THEN
                     DO K1 = 1,NDOFEL
                          DO K2 = 1,NDOFEL
                               AMATRX(K2,K1) = SPRINGEK(K2,K1)
                          ENDDO
                     ENDDO
                  ENDIF
            ENDIF
            IF(LFLAGS(3).EQ.4) THEN
                  IF(JTYPE.EQ.1.OR.JTYPE.EQ.2) THEN
                     DO K1 = 1,3
                          AMATRX(K1,K1) = MASS
                     ENDDO
                  ENDIF
                  IF(JTYPE.EQ.3.OR.JTYPE.EQ.4.OR.JTYPE.EQ.5) THEN
                     DO K1 = 1,NDOFEL
                        AMATRX(K1,K1) = MASS
                     ENDDO
                  ENDIF
            ENDIF
      ENDIF

      ITIME = ITIME + 1
C
C  PORT NUMBER 7 IS USED TO WRITE DATA INTO MSG FILE
C  USED ONLY FOR DEBUG
C  FOR DEBUG PURPOSE
C       WRITE(7,*) 'ICALL=',ITIME
C  WRTING AFTER EVERY ITERATION, I.E., ITIME.EQ.NTOTEL
C  AND REWINDING
C
      IF(ITIME.EQ.NTOTEL) THEN
         WRITE(105,*) ETTOT + ESPR,espr,itime
         ITIME = 0
         REWIND(2000)
         REWIND(6000)
      ENDIF
      RETURN
```

```
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  P R O G R A M                                                     .
C .       TO CALCULATE                                                 .
C .            SPRING STIFFNESS MATRIX                                 .
C .            SPRING RESIDUAL FORCE VECTOR                            .
C .       OF SPRNIG ELEMNTS (TYPE III IV, AND V)                       .
C .                                                                    .
C .  N O T E                                                           .
C .        INPUT  : SPRINGEX, SPRINGEX0, NEE, JTYPE                    .
C .        OUTPUT : SPRINGEK, SPRINGEF, ESPR                           .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

      SUBROUTINE SPRINGEMEF(SPRINGEX,SPRINGEX0,SPRINGEK,SPRINGEF,ESPR,
     &                      NEE,JTYPE)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 SPRINGEX(2,3),SPRINGEX0(2,3),SPRINGEK(6,6),SPRINGEF(6)
      DIMENSION R0REF(3)
C
C  THE THREE BOND LENGHTS IF ELEMENTS III, IV, V RESPECTIVELY
C
      R0REF(1) = 1.D0
      R0REF(2) = DSQRT(3.D0)/3.D0*R0REF(1)
      R0REF(3) = R0REF(1)*2.D0
      STIFF = 1.0D7

      X1 = SPRINGEX(1,1)
      Y1 = SPRINGEX(1,2)
      Z1 = SPRINGEX(1,3)
      X2 = SPRINGEX(2,1)
      Y2 = SPRINGEX(2,2)
      Z2 = SPRINGEX(2,3)

      R = DSQRT((X2-X1)**2.D0 + (Y2-Y1)**2.D0 + (Z2-Z1)**2.D0)
      R0 = R0REF(JTYPE-2)
C
C  FOR DEBUG PURPOSE
C     IF(DABS(R-R0).GE.1D-3) THEN
C           WRITE(7,743) NEE, R, R0, R - R0
C743        FORMAT(i5,',',3f20.15)
C     ENDIF
C
      ESPR = ESPR + 0.5D0*STIFF*(R-R0)**2.D0
      DR1 = STIFF*(1.D0-R0/R)
      DDR2 = STIFF * R0/R**3.D0
C
C  CONSTRUCTING THE SPRING STIFFNESS MATRIX AND RESIDUAL FORCE VECTOR
C
      SPRINGEF(1) = DR1*(X1-X2)
      SPRINGEF(2) = DR1*(Y1-Y2)
      SPRINGEF(3) = DR1*(Z1-Z2)
      SPRINGEF(4) = -DR1*(X1-X2)
      SPRINGEF(5) = -DR1*(Y1-Y2)
      SPRINGEF(6) = -DR1*(Z1-Z2)
```

```
      SPRINGEK(1,1) = DDR2*(X1-X2)*(X1-X2)+DR1
      SPRINGEK(1,2) = DDR2*(X1-X2)*(Y1-Y2)
      SPRINGEK(1,3) = DDR2*(X1-X2)*(Z1-Z2)
      SPRINGEK(1,4) = -SPRINGEK(1,1)
      SPRINGEK(1,5) = -SPRINGEK(1,2)
      SPRINGEK(1,6) = -SPRINGEK(1,3)

      SPRINGEK(2,1) = SPRINGEK(1,2)
      SPRINGEK(2,2) = DDR2*(Y1-Y2)*(Y1-Y2)+DR1
      SPRINGEK(2,3) = DDR2*(Y1-Y2)*(Z1-Z2)
      SPRINGEK(2,4) = -SPRINGEK(2,1)
      SPRINGEK(2,5) = -SPRINGEK(2,2)
      SPRINGEK(2,6) = -SPRINGEK(2,3)

      SPRINGEK(3,1) = SPRINGEK(1,3)
      SPRINGEK(3,2) = SPRINGEK(2,3)
      SPRINGEK(3,3) = DDR2*(Z1-Z2)*(Z1-Z2)+DR1
      SPRINGEK(3,4) = -SPRINGEK(3,1)
      SPRINGEK(3,5) = -SPRINGEK(3,2)
      SPRINGEK(3,6) = -SPRINGEK(3,3)

      SPRINGEK(4,1) = SPRINGEK(1,4)
      SPRINGEK(4,2) = SPRINGEK(2,4)
      SPRINGEK(4,3) = SPRINGEK(3,4)
      SPRINGEK(4,4) = -SPRINGEK(4,1)
      SPRINGEK(4,5) = -SPRINGEK(4,2)
      SPRINGEK(4,6) = -SPRINGEK(4,3)

      SPRINGEK(5,1) = SPRINGEK(1,5)
      SPRINGEK(5,2) = SPRINGEK(2,5)
      SPRINGEK(5,3) = SPRINGEK(3,5)
      SPRINGEK(5,4) = -SPRINGEK(5,1)
      SPRINGEK(5,5) = -SPRINGEK(5,2)
      SPRINGEK(5,6) = -SPRINGEK(5,3)

      SPRINGEK(6,1) = SPRINGEK(1,6)
      SPRINGEK(6,2) = SPRINGEK(2,6)
      SPRINGEK(6,3) = SPRINGEK(3,6)
      SPRINGEK(6,4) = -SPRINGEK(6,1)
      SPRINGEK(6,5) = -SPRINGEK(6,2)
      SPRINGEK(6,6) = -SPRINGEK(6,3)
      END

C . . . . . .
C .                                                                    .
C .  P R O G R A M                                                     .
C .       TO CALCULATE                                                 .
C .            ELEMENTAL STIFFNESS MATRIX                              .
C .            ELEMENTAL RESIDUAL VECTOR                               .
C .       OF TYPE I AND II ELEMENTS                                    .
C .                                                                    .
C .  N O T E                                                           .
C .        INPUT  : EX, NELE, JTYPE, JELEM                             .
C .        OUTPUT : EK, EF, ETOT1                                      .
C .        STIFFNESS MATRIX, EK :                                      .
C .              IT IS SYMMETRIC                                       .
```

```
C .              IT OBTAINED USING,                                      .
C .              0.5 Partial V / (Partial (X(1)_L) Partial (X(I)_J))    .
C .              THE COEFFICENT 0.5 IS USED TO AVOID DOUBLE COUNT OF    .
C .              OF ENERGY FROM THE SAME TETRA HEDERAL ANGLE.           .
C .              ITS SIZE IS 57X57                                      .
C .         UNEQUILIBRIUM FORCE VECTOR, EF :                            .
C .              IT OBTAINED USING,                                     .
C .              Partial V / (Partial (X(1)_L) AT THE CENTRE VERTEX     .
C .              ITS SIZE IS 57X1                                       .
C .                                                                     .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      SUBROUTINE CALEMEF(EX,NELE,EK,EF,ETTOT1,JTYPE)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 EX(19,3),EK(57,57),EF(57)
      REAL*8 X(3,3),PVPX(3),PPVPXPX(3,3,3)
      INTEGER NELE(19),NN(3)
      DIMENSION ASTIFF(9),AE(9)
      DIMENSION NTA(2),NANGLE(2,9),NAORDER(2,9,18)
      DIMENSION TSTIFF(3)
C
C  STIFFNESSES FOR THE 3 TYPES OF DIHEDERAL ANGLE
C
      TSTIFF = 0.D0
      TSTIFF(1) = 1.D2
      TSTIFF(2) = 1.D2
      TSTIFF(3) = 1.D2
C
C  ASSIGNING STIFFENESS TO THE 9 ANGLES IN AN ELEMENT
C
      ASTIFF(1) = TSTIFF(1)
      ASTIFF(2) = TSTIFF(2)
      ASTIFF(3) = 0.D0
      ASTIFF(4) = TSTIFF(3)
      ASTIFF(5) = 0.D0
      ASTIFF(6) = 0.D0
      ASTIFF(7) = 0.D0
      ASTIFF(8) = 0.D0
      ASTIFF(9) = 0.D0
C
C  EQUILIBRUM VALUES OF ETA, THETA AND 3 DIHEDERAL ANGLES
C
      PI = DACOS(-1.D0)
      THETA = PI/6.D0
      ETA= 0 .158964016016962990 04D0
      ALPHA1 = 1.088176213364169 0818D0
      ALPHA2 = 2.541879510749498 4728D0
      ALPHA3 = 1.894524242086 61D0
C
C  ASSIGNING EQUILIBRIUM ANGLE VALUES DIHEDERAL ANGLES TO THE 9 ANGLES
C  IN AN ELEMENT
C
      AE(1) = ALPHA1
      AE(2) = ALPHA2
      AE(3) = 0.D0
      AE(4) = ALPHA3
      AE(5) = 0.D0
      AE(6) = 0.D0
```

```
AE(7) = 0.D0
AE(8) = 0.D0
AE(9) = 0.D0

NTA(1) = 6
NTA(2) = 9

NANGLE = 0
NAORDER = 0

NANGLE(1,1) = 3
NANGLE(1,2) = 3
NANGLE(1,3) = 3
NANGLE(1,4) = 6
NANGLE(1,5) = 6
NANGLE(1,6) = 6

NANGLE(2,1) = 2
NANGLE(2,2) = 2
NANGLE(2,3) = 2
NANGLE(2,4) = 4
NANGLE(2,5) = 1
NANGLE(2,6) = 4
NANGLE(2,7) = 3
NANGLE(2,8) = 6
NANGLE(2,9) = 3

NAORDER(1,1,1) = 2
NAORDER(1,1,2) = 1
NAORDER(1,1,3) = 3
NAORDER(1,1,4) = 2
NAORDER(1,1,5) = 1
NAORDER(1,1,6) = 4
NAORDER(1,1,7) = 3
NAORDER(1,1,8) = 1
NAORDER(1,1,9) = 4


NAORDER(1,2,1) = 5
NAORDER(1,2,2) = 1
NAORDER(1,2,3) = 6
NAORDER(1,2,4) = 5
NAORDER(1,2,5) = 1
NAORDER(1,2,6) = 7
NAORDER(1,2,7) = 6
NAORDER(1,2,8) = 1
NAORDER(1,2,9) = 7

NAORDER(1,3,1) = 2
NAORDER(1,3,2) = 1
NAORDER(1,3,3) = 7
NAORDER(1,3,4) = 3
NAORDER(1,3,5) = 1
NAORDER(1,3,6) = 6
NAORDER(1,3,7) = 4
NAORDER(1,3,8) = 1
NAORDER(1,3,9) = 5
```

```
NAORDER(1,4,1) = 1
NAORDER(1,4,2) = 2
NAORDER(1,4,3) = 8
NAORDER(1,4,4) = 1
NAORDER(1,4,5) = 2
NAORDER(1,4,6) = 10
NAORDER(1,4,7) = 1
NAORDER(1,4,8) = 3
NAORDER(1,4,9) = 11
NAORDER(1,4,10) = 1
NAORDER(1,4,11) = 3
NAORDER(1,4,12) = 13
NAORDER(1,4,13) = 1
NAORDER(1,4,14) = 4
NAORDER(1,4,15) = 14
NAORDER(1,4,16) = 1
NAORDER(1,4,17) = 4
NAORDER(1,4,18) = 16

NAORDER(1,5,1) = 1
NAORDER(1,5,2) = 2
NAORDER(1,5,3) = 9
NAORDER(1,5,4) = 1
NAORDER(1,5,5) = 3
NAORDER(1,5,6) = 12
NAORDER(1,5,7) = 1
NAORDER(1,5,8) = 4
NAORDER(1,5,9) = 15
NAORDER(1,5,10) = 1
NAORDER(1,5,11) = 5
NAORDER(1,5,12) = 17
NAORDER(1,5,13) = 1
NAORDER(1,5,14) = 6
NAORDER(1,5,15) = 18
NAORDER(1,5,16) = 1
NAORDER(1,5,17) = 7
NAORDER(1,5,18) = 19

NAORDER(1,6,1) = 1
NAORDER(1,6,2) = 5
NAORDER(1,6,3) = 8
NAORDER(1,6,4) = 1
NAORDER(1,6,5) = 5
NAORDER(1,6,6) = 11
NAORDER(1,6,7) = 1
NAORDER(1,6,8) = 6
NAORDER(1,6,9) = 10
NAORDER(1,6,10) = 1
NAORDER(1,6,11) = 6
NAORDER(1,6,12) = 14
NAORDER(1,6,13) = 1
NAORDER(1,6,14) = 7
NAORDER(1,6,15) = 13
NAORDER(1,6,16) = 1
NAORDER(1,6,17) = 7
NAORDER(1,6,18) = 16
```

```
NAORDER(2,1,1) = 1
NAORDER(2,1,2) = 2
NAORDER(2,1,3) = 8
NAORDER(2,1,4) = 1
NAORDER(2,1,5) = 2
NAORDER(2,1,6) = 10

NAORDER(2,2,1) = 1
NAORDER(2,2,2) = 3
NAORDER(2,2,3) = 11
NAORDER(2,2,4) = 1
NAORDER(2,2,5) = 3
NAORDER(2,2,6) = 13

NAORDER(2,3,1) = 1
NAORDER(2,3,2) = 2
NAORDER(2,3,3) = 9
NAORDER(2,3,4) = 1
NAORDER(2,3,5) = 3
NAORDER(2,3,6) = 12

NAORDER(2,4,1) = 2
NAORDER(2,4,2) = 1
NAORDER(2,4,3) = 6
NAORDER(2,4,4) = 2
NAORDER(2,4,5) = 1
NAORDER(2,4,6) = 7
NAORDER(2,4,7) = 1
NAORDER(2,4,8) = 4
NAORDER(2,4,9) = 15
NAORDER(2,4,10) = 1
NAORDER(2,4,11) = 5
NAORDER(2,4,12) = 17

NAORDER(2,5,1) =2
NAORDER(2,5,2) =1
NAORDER(2,5,3) =3

NAORDER(2,6,1) = 3
NAORDER(2,6,2) = 1
NAORDER(2,6,3) = 4
NAORDER(2,6,4) = 3
NAORDER(2,6,5) = 1
NAORDER(2,6,6) = 5
NAORDER(2,6,7) = 1
NAORDER(2,6,8) = 6
NAORDER(2,6,9) = 15
NAORDER(2,6,10) = 1
NAORDER(2,6,11) = 7
NAORDER(2,6,12) = 17

NAORDER(2,7,1) = 4
NAORDER(2,7,2) = 1
NAORDER(2,7,3) = 5
NAORDER(2,7,4) = 1
NAORDER(2,7,5) = 6
```

```
      NAORDER(2,7,6) = 11
      NAORDER(2,7,7) = 1
      NAORDER(2,7,8) = 7
      NAORDER(2,7,9) = 13

      NAORDER(2,8,1) = 4
      NAORDER(2,8,2) = 1
      NAORDER(2,8,3) = 7
      NAORDER(2,8,4) = 5
      NAORDER(2,8,5) = 1
      NAORDER(2,8,6) = 6
      NAORDER(2,8,7) = 1
      NAORDER(2,8,8) = 4
      NAORDER(2,8,9) = 14
      NAORDER(2,8,10) = 1
      NAORDER(2,8,11) = 5
      NAORDER(2,8,12) = 16
      NAORDER(2,8,13) = 1
      NAORDER(2,8,14) = 6
      NAORDER(2,8,15) = 18
      NAORDER(2,8,16) = 1
      NAORDER(2,8,17) = 7
      NAORDER(2,8,18) = 19

      NAORDER(2,9,1) = 6
      NAORDER(2,9,2) = 1
      NAORDER(2,9,3) = 7
      NAORDER(2,9,4) = 1
      NAORDER(2,9,5) = 4
      NAORDER(2,9,6) = 8
      NAORDER(2,9,7) = 1
      NAORDER(2,9,8) = 5
      NAORDER(2,9,9) = 10

      EK = 0.D0
      EF = 0.D0
C
C  CONSTRUCTING THE ELEMENTAL STIFFNESS MATRIX AND RESIDUAL FORCE
C  VECTOR
C
      DO IANGLE = 1,NTA(JTYPE)
         NLOOP = NANGLE(JTYPE,IANGLE)
         DO ILOOP = 1,NLOOP
            I1 = NAORDER(JTYPE,IANGLE,(ILOOP-1)*3 + 1)
            I2 = NAORDER(JTYPE,IANGLE,(ILOOP-1)*3 + 2)
            I3 = NAORDER(JTYPE,IANGLE,(ILOOP-1)*3 + 3)
            IF((NELE(I1).NE.1.AND.NELE(I3).NE.1).OR.(NELE(I2).NE.1.AND.
     &          NELE(I3).NE.1)) THEN
               NN(1) = I1
               NN(2) = I2
               NN(3) = I3
               STIFF = ASTIFF(IANGLE)
               EANGLE = AE(IANGLE)
               IF(IANGLE.EQ.3) NN = 1
                     IF(IANGLE.EQ.5) NN = 1
                     IF(IANGLE.EQ.6) NN = 1
                     IF(IANGLE.EQ.7) NN = 1
```

```
                    IF(IANGLE.EQ.8) NN = 1

                    IF(IANGLE.EQ.9) NN = 1
               ELSE
                  NN = 1
               ENDIF
               IF(NN(1) + NN(2) + NN(3).NE.3) THEN
                  DO II = 1,3
                     X(II,:) = EX(NN(II),:)
                  ENDDO
C
C  CALCULATING THE PARTIAL DERIVATES OF ENERGY (V)
C
               CALL DIFF(IANGLE,STIFF,EANGLE,X,NN,PVPX,PPVPXPX,ET,
     &                   JTYPE,JELEM)
               IF(NN(2).EQ.1) ETTOT1 = ETTOT1 + ET
               DO L = 1,3
                   EF(L) = EF(L)+PVPX(L)
               ENDDO
               DO L = 1,3
                  DO I = 1,3
                     DO J = 1,3
                        IF(NN(I).EQ.1) THEN
                           EK(L,J) = EK(L,J) + PPVPXPX(L,I,J)
                        ELSE
                           EK(L,3*(NN(I)-1)+J) = EK(L,3*(NN(I)-1)+J)
     &                                         + 0.5*PPVPXPX(L,I,J)
                           EK(3*(NN(I)-1)+J,L) = EK(L,3*(NN(I)-1)+J)
                        ENDIF
                     ENDDO
                  ENDDO
               ENDDO
            ENDIF
         ENDDO
      ENDDO
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  P R O G R A M                                                     .
C .       TO CALCULATE FIRST TWO DERIVATIVE OF STAIN ENERGY            .
C .                                                                    .
C .  N O T E                                                           .
C .        INPUT  : IANGLE, STIFF, EANGLE, X, NSE, JTYPE, JELEM       .
C .        OUTPUT : PVPX, PPVPXPX, ET                                  .
C .        REFER THESIS FOR THE FORMULAS USED IN THIS SECTION          .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      SUBROUTINE DIFF(IANGLE,STIFF,EANGLE,X,NSE,PVPX,PPVPXPX,ET
     &                JTYPE,JELEM)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 X(3,3),R(3,3),PCPX(3,3),PVPX(3),PRPX(3,3,3,3)
      REAL*8 PPCPXPX(3,3,3,3),PPVPXPX(3,3,3)
      INTEGER NSE(3)

      PI = DACOS(-1.D0)
```

```
      ET = 0.D0

      DO I = 1,3
         DO J = 1,3
            R(I,J) = DISTANCE(X(I,:),X(J,:))
         ENDDO
      ENDDO

      COS_THETA = CFC(R(1,2),R(2,3),R(1,3))
C
C  THE FOLLWING STEP IS LOGICAL LIMIT, TO PREVENT THE PROGRAM FROM
C  CRASHING. NUMERICAL ROUND OFF SOMETIMES MAKES COS AND SIN > 1
C
      IF(DABS(DABS(COS_THETA) - 1.D0).LE.1D-3) THEN
         IF(DABS(COS_THETA - 1.D0).LE.1D-3) THETA = 0.D0
         IF(DABS(COS_ALPHA + 1.D0).LE.1D-3) THETA = PI
         SIN_THETA = 0.01D0
      ELSE
         THETA = DACOS(COS_THETA)
         SIN_THETA = DSQRT(1.D0 - COS_THETA**2.D0)
      ENDIF

      IF(IANGLE.EQ.1.OR.IANGLE.EQ.2) THEN
         FACTOR1 = 4.D0/3.D0
         FACTOR2 = -1.D0/3.D0
      ENDIF
      IF(IANGLE.EQ.4) THEN
         FACTOR1 = 2.D0/DSQRT(3.D0)
         FACTOR2 = 0.D0
      ENDIF

      COS_ALPHA = FACTOR1*COS_THETA + FACTOR2
C
C  LOGICAL LIMIT
C
      IF(DABS(DABS(COS_ALPHA) - 1.D0).LE.1D-3.OR.
     &   DABS(COS_ALPHA).GT.1.D0) THEN
         SIN_ALPHA = 0.01D0
         IF(DABS(COS_ALPHA - 1.D0).LE.1D-3) ALPHA = 0.D0
         IF(DABS(COS_ALPHA + 1.D0).LE.1D-3) ALPHA = PI
         IF(COS_ALPHA.LT.-1.D0) ALPHA = PI
         IF(COS_ALPHA.GT.1.D0) ALPHA = 0.D0
      ELSE
         SIN_ALPHA = DSQRT(1.D0 - COS_ALPHA**2.D0)
         ALPHA = DACOS(COS_ALPHA)
      ENDIF

      PVPA = STIFF*(ALPHA - EANGLE)
      PPVPAPA = STIFF

      PTPC = -1.D0/SIN_THETA
      PPTPCPC = -COS_THETA/SIN_THETA**3.D0

      PAPT = FACTOR1*SIN_THETA/SIN_ALPHA
      PPAPTPT = FACTOR1*(COS_THETA/SIN_ALPHA - SIN_THETA*COS_ALPHA
     &      /SIN_ALPHA**2.D0*PAPT)
```

```
      IF(JTYPE.EQ.1) THEN
            ET = 0.5D0*STIFF*(ALPHA - EANGLE)**2.D0
      ENDIF

      DO I = 1,3
         DO J = 1,3
            PCPX(I,J) =
     &        PCPR1FC(R(1,2),R(2,3),R(1,3))
     &      * PRPXFC(1,2,I,J,X(1,:),X(2,:),R(1,2))
     &      + PCPR2FC(R(1,2),R(2,3),R(1,3))
     &      * PRPXFC(2,3,I,J,X(2,:),X(3,:),R(2,3))
     &      + PCPR3FC(R(1,2),R(2,3),R(1,3))
     &      * PRPXFC(1,3,I,J,X(1,:),X(3,:),R(1,3))
         ENDDO
      ENDDO

      IF(NSE(1).EQ.1) PVPX = PVPA*PAPT*PTPC*PCPX(1,:)
      IF(NSE(2).EQ.1) PVPX = PVPA*PAPT*PTPC*PCPX(2,:)


      PCPR12 = PCPR1FC(R(1,2),R(2,3),R(1,3))
      PCPR23 = PCPR2FC(R(1,2),R(2,3),R(1,3))
      PCPR13 = PCPR3FC(R(1,2),R(2,3),R(1,3))
      PPCPR12PR12 = PPCPRPR11FC(R(1,2),R(2,3),R(1,3))
      PPCPR12PR23 = PPCPRPR12FC(R(1,2),R(2,3),R(1,3))
      PPCPR12PR13 = PPCPRPR13FC(R(1,2),R(2,3),R(1,3))
      PPCPR23PR23 = PPCPRPR22FC(R(1,2),R(2,3),R(1,3))
      PPCPR23PR13 = PPCPRPR23FC(R(1,2),R(2,3),R(1,3))
      PPCPR13PR13 = PPCPRPR33FC(R(1,2),R(2,3),R(1,3))

      PRPX = 0.
      DO I = 1,3
         DO J = 1,3
            DO K = 1,3
               DO L = 1,3
                  PRPX(I,J,K,L) = PRPXFC(I,J,K,L,X(I,:),X(J,:),R(I,J))
               ENDDO
            ENDDO
         ENDDO
      ENDDO

      DO I = 1,3
         DO J = 1,3
            DO K = 1,3
               DO L = 1,3
           PPCPXPX(I,J,K,L) = (PPCPR12PR12*PRPX(1,2,K,L) + PPCPR12PR23
     &             *PRPX(2,3,K,L) +
PPCPR12PR13*PRPX(1,3,K,L))*PRPX(1,2,I,J)
     &        + PCPR12*PPRPXPXFC(1,2,I,K,J,L,X(1,:),X(2,:),R(1,2))
     &        + (PPCPR12PR23*PRPX(1,2,K,L)+PPCPR23PR23
     &            *PRPX(2,3,K,L) +
PPCPR23PR13*PRPX(1,3,K,L))*PRPX(2,3,I,J)
     &        + PCPR23*PPRPXPXFC(2,3,I,K,J,L,X(2,:),X(3,:),R(2,3))
     &        + (PPCPR12PR13*PRPX(1,2,K,L) + PPCPR23PR13
     &            *PRPX(2,3,K,L) +
PPCPR13PR13*PRPX(1,3,K,L))*PRPX(1,3,I,J)
     &        + PCPR13*PPRPXPXFC(1,3,I,K,J,L,X(1,:),X(3,:),R(1,3))
```

```
                ENDDO
              ENDDO
            ENDDO
          ENDDO

          IF(NSE(2).EQ.1) M = 2
          IF(NSE(1).EQ.1) M = 1

          DO J = 1,3
             DO K = 1,3
               DO L = 1,3
            PPVPXPX(J,K,L) = PPVPAPA*PAPT**2.D0*PTPC**2.D0*PCPX(M,J)*
     &      PCPX(K,L) + PVPA*PPAPTPT*PTPC**2.D0*PCPX(M,J)*PCPX(K,L)
     &        + PVPA*PAPT*PPTPCPC*PCPX(M,J)*PCPX(K,L)
     &          + PVPA*PAPT*PTPC*PPCPXPX(M,J,K,L)
               ENDDO
             ENDDO
          ENDDO

          RETURN
          END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  F U N C T I O N                                                   .
C .       TO CALCULATE DISTANE BETWEEN TWO POINTS                      .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
          FUNCTION DISTANCE(X1,X2)
          IMPLICIT DOUBLE PRECISION (A-H, O-Z)
          REAL*8 DISTANCE
          REAL*8 X1(3),X2(3)
          DISTANCE = DSQRT((X1(1)-X2(1))**2.D0 + (X1(2)-X2(2))**2.D0
     &            + (X1(3)-X2(3))**2.D0)
          RETURN
          END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  F U N C T I O N                                                   .
C .       TO CALCULATE COS ANGLE FORMED BY B, AND C                    .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

          FUNCTION CFC(B,C,A)
          IMPLICIT DOUBLE PRECISION (A-H, O-Z)
          REAL*8 CFC
          CFC = (B*B + C*C - A*A)/(2.D0*B*C)
          RETURN
          END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  F U N C T I O N                                                   .
C .       TO CALCULATE FIRST DERIVATIVE OF COS FUNCTION                .
C .                                                                    .
C .  N O T E                                                           .
```

```
C .        COS(ANGLE) = (B^2+C^2-A^2/(2*B*C)                          .
C .        THE DERIVATIVE IS WITH REPECT TO B                         .
C .                                                                   .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PCPR1FC(B,C,A)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PCPR1FC
      PCPR1FC = 0.5*(B**2.D0 - C**2.D0 + A**2.D0)/B**2.D0/C
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                   .
C .  F U N C T I O N                                                  .
C .       TO CALCULATE FIRST DERIVATIVE OF COS FUNCTION               .
C .                                                                   .
C .  N O T E                                                          .
C .        COS(ANGLE) = (B^2+C^2-A^2/(2*B*C)                          .
C .        THE DERIVATIVE IS WITH REPECT TO C                         .
C .                                                                   .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PCPR2FC(B,C,A)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PCPR2FC
      PCPR2FC = 0.5*(C**2.D0 - B**2.D0 + A**2.D0)/B/C**2.D0
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                   .
C .  F U N C T I O N                                                  .
C .       TO CALCULATE FIRST DERIVATIVE OF COS FUNCTION               .
C .                                                                   .
C .  N O T E                                                          .
C .        COS(ANGLE) = (B^2+C^2-A^2/(2*B*C)                          .
C .        THE DERIVATIVE IS WITH REPECT TO A                         .
C .                                                                   .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

      FUNCTION PCPR3FC(B,C,A)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PCPR3FC
      PCPR3FC = -A/B/C
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                   .
C .  F U N C T I O N                                                  .
C .       TO CALCULATE SECOND DERIVATIVE OF COS FUNCTION              .
C .                                                                   .
C .  N O T E                                                          .
C .        COS(ANGLE) = (B^2+C^2-A^2/(2*B*C)                          .
C .        THE DERIVATIVE IS WITH REPECT TO BB                        .
C .                                                                   .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PPCPRPR11FC(B,C,A)
```

```
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PPCPRPR11FC
      PPCPRPR11FC = (C**2.D0 - A**2.D0)/(B**3.D0*C)
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  F U N C T I O N                                                   .
C .       TO CALCULATE SECOND DERIVATIVE OF COS FUNCTION               .
C .                                                                    .
C .  N O T E                                                           .
C .        COS(ANGLE) = (B^2+C^2-A^2/(2*B*C)                           .
C .        THE DERIVATIVE IS WITH REPECT TO BC                         .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PPCPRPR12FC(B,C,A)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PPCPRPR12FC
      PPCPRPR12FC = -(B**2.D0+C**2.D0+A**2.D0)/(2.D0*B**2.D0*C**2.D0)
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  F U N C T I O N                                                   .
C .       TO CALCULATE SECOND DERIVATIVE OF COS FUNCTION               .
C .                                                                    .
C .  N O T E                                                           .
C .        COS(ANGLE) = (B^2+C^2-A^2/(2*B*C)                           .
C .        THE DERIVATIVE IS WITH REPECT TO BA                         .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PPCPRPR13FC(B,C,A)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PPCPRPR13FC
      PPCPRPR13FC = A/(B**2.D0*C)
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  F U N C T I O N                                                   .
C .       TO CALCULATE SECOND DERIVATIVE OF COS FUNCTION               .
C .                                                                    .
C .  N O T E                                                           .
C .        COS(ANGLE) = (B^2+C^2-A^2/(2*B*C)                           .
C .        THE DERIVATIVE IS WITH REPECT TO CC                         .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PPCPRPR22FC(B,C,A)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PPCPRPR22FC
      PPCPRPR22FC = (B**2.D0 - A**2.D0)/(B*C**3.D0)
      RETURN
      END
```

```
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                   .
C .  F U N C T I O N                                                  .
C .       TO CALCULATE SECOND DERIVATIVE OF COS FUNCTION              .
C .                                                                   .
C .  N O T E                                                          .
C .        COS(ANGLE) = (B^2+C^2-A^2/(2*B*C)                          .
C .        THE DERIVATIVE IS WITH REPECT TO CA                        .
C .                                                                   .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PPCPRPR23FC(B,C,A)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PPCPRPR23FC
      PPCPRPR23FC = A/(B*C**2.D0)
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                   .
C .  F U N C T I O N                                                  .
C .       TO CALCULATE SECOND DERIVATIVE OF COS FUNCTION              .
C .                                                                   .
C .  N O T E                                                          .
C .        COS(ANGLE) = (B^2+C^2-A^2/(2*B*C)                          .
C .        THE DERIVATIVE IS WITH REPECT TO AA                        .
C .                                                                   .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PPCPRPR33FC(B,C,A)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PPCPRPR33FC
      PPCPRPR33FC = -1.D0/(B*C)
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                   .
C .  F U N C T I O N                                                  .
C .       TO CALCULATE Partial R / Partial X                          .
C .                                                                   .
C .  N O T E                                                          .
C .        R : The distance of atom I and J                           .
C .        K : If K=I, the derivative is respect to atom I            .
C .          : If K=J, the derivative is respect to atom J            .
C .        L : Direction of the derivative. 1, 2, and 3 means         .
C .            X, Y, and Z                                            .
C .                                                                   .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PRPXFC(I,J,K,L,XI,XJ,R)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PRPXFC,XI(3),XJ(3)
      IF((I.EQ.J).OR.((K.NE.I).AND.(K.NE.J))) THEN
         PRPXFC = 0.
         RETURN
      ENDIF
      IF(K.EQ.I) THEN
         PRPXFC = (XI(L) - XJ(L))/R
         RETURN
```

```
      ENDIF
      IF(K.EQ.J) THEN
         PRPXFC = (XJ(L) - XI(L))/R
         RETURN
      ENDIF
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C .  F U N C T I O N                                                   .
C .       TO CALCULATE                                                 .
C .       Partial (R_IJ) / Partial (X(K)_M) Partial (X(L)_N)           .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      FUNCTION PPRPXPXFC(I,J,K,L,M,N,XK,XL,RIJ)
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
      REAL*8 PPRPXPXFC,XK(3),XL(3)
      IF ((K.NE.I.AND.K.NE.J).OR.(L.NE.I.AND.L.NE.J)) THEN
          PPRPXPXFC = 0.D0
      RETURN
      ENDIF
      IF(K.EQ.L) THEN
         IF(M.EQ.N) THEN
            PPRPXPXFC = 1.D0/RIJ - (XK(M) - XL(M))**2.D0/RIJ**3.D0
            RETURN
         ENDIF
         IF(M.NE.N) THEN
            PPRPXPXFC = -(XK(M) - XL(M))*(XK(N) - XL(N))/RIJ**3.D0
            RETURN
         ENDIF
      ENDIF
      IF(K.NE.L) THEN
         IF(M.EQ.N) THEN
            PPRPXPXFC = -1.D0/RIJ + (XK(M) - XL(M))**2.D0/RIJ**3.D0
            RETURN
         ENDIF
         IF(M.NE.N) THEN
            PPRPXPXFC = (XK(M) - XL(M))*(XK(N) - XL(N))/RIJ**3.D0
            RETURN
         ENDIF
      ENDIF
      PPRPXPXFC = 0.D0
      RETURN
      END

C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C .                                                                    .
C . P A R A M E T E R S   U S E D   A N D   T H I E R   M E A N I N G .
C .                                                                    .
C . AE         : ARRAY CONTAINING THE EQUILIBRIUM VALUES OF THE 9      .
C .              DIHEDERAL                                             .
C .              ANGLES IN THE ELEMENT                                 .
C . AMATRX     : JACOBIAN MATRIX, DEPENDING ON THE SIMULATION IT CAN   .
C .              BE K OR M                                             .
C .              MATRIX                                                .
C . ASTIFF     : ARRAY CONTAINING THE STIFFNESS VALUES OF THE 9        .
```

```
C .              DIHEDERAL ANGLES                                        .
C .              IN THE ELEMENT                                          .
C . COORDS     : ARRAY CONTAINING THE ORIGINAL COORDINATES OF THE        .
C .              ELEMENT                                                 .
C . DR1        : FIRST DERIVATIVE WRT R                                  .
C . DDR2       : SECOND DERIVATIVE WRT R                                 .
C . EF         : ELEMENTAL UNBALCAED FORCE VECTOR                        .
C . EK         : ELEMENTAL STIFFNESS MATRIX                              .
C . ESPR       : TOTAL ENERGY OF SPRING ELEMENTS                         .
C . ETTOT      : TOTAL ENERGY IN THE 19 NODE ELEMENTS                    .
C . EX         : ARRAY CONTAINING THE CURRENT COORDINATES OF THE         .
C .              ELEMENT                                                 .
C . IMCRD      : MAXIMUM NUMBER OF COORDINATES NEEDED TO REPRESENT       .
C .              A NODE (VALUE IS 3)                                     .
C . ITIME      : COUNTER THAT KEEPS TRACK OF NUMBER OF ELEMENTS          .
C .              SIMULATED IN AN ITERATION                               .
C . JTYPE      : THE TYPE OF ELEMENT IN THE CURRENT LOOP                 .
C . JELEM      : ELEMENT NUMBER ASSIGNED TO THE CURRENT ELEMENT          .
C . LFLAGS     : LFLAGS(1) = 1  -> STATIC ANALYSIS                       .
C .              LFLAGS(1) = 41 -> EIGEN FREQUENCY ANALYSIS              .
C .              LFLAGS(1) = 2  -> STATIC ANALYSIS                       .
C .              LFLAGS(3) = 2  -> DEFINES CURRENT STIFFNESS MATRIX      .
C .              LFLAGS(3) = 4  -> DEFINES CURRENT MASS MATRIX           .
C . MASS       : MASS MATRIX                                             .
C . NANGLE     : ARRAY TO STORE THE NUMBER OF ANGLES IN EACH TYPE FOR    .
C .              TYPE I AND II ELEMENTS                                  .
C . NAORDER    : ARRAY TO SOTRE THE NODE NUMER OF ANGLES IN NANGLE       .
C . NDOFEL     : NUMBER OF DEGREES OF FREEDOM FOR THE ELEMENT            .
C .              (VALUE = 57)                                            .
C . NELE       : ARRAY STORING THE GLOBAL ELEMENT NODE NUMBERS           .
C . NN         : ARRAY CONTAINING THE NODE NUMBERS OF THE 3 NODES        .
C .              CONTRIBUTING AN ANGLE UNDER CONSIDERATION               .
C . NNODE      : NUMBER OF NODES IN THE ELEMENT                          .
C .              (VALUE = 19)                                            .
C . NTA        : ARRAY TO STORE NUMBER OF TYPES OF ANGLES IN THE TYPE    .
C .              I AND II ELEMENTS                                       .
C . NTOTEL     : TOTAL NUMBER OF ELEMENTS                                .
C . PVPA       : PARTIAL DEIVATIVE OF ENERGY WRT TO A. USING THE SAME    .
C .              CONVETION THE NOTATIONS SUCH AS, PPVPAPA, PCPX, ETC.    .
C .              CAN BE INFERRED                                         .
C . PPVPXPX    : SECOND DERIVATIV.E OF ENERGY(V) WRT POSITOIN OF THE     .
C .              CENTRAL ATOM(X)                                         .
C . PVPX       : FIRST DERIVATIVE OF ENERGY(V) WRT POSITOIN OF THE       .
C .              CENTRAL ATOM(X)                                         .
C . R          : LENGTH OF BOND BEFORE DEFORMATION                       .
C . RHS        : RIGHT HAND SIDE RESIDUAL VECTOR                         .
C . R0         : ORIGINAL LENGTH OF BOND                                 .
C . R0REF      : ARRAY STROING THE BOND LENTHS OF ELEMENTS III, IV,      .
C .              AND V                                                   .
C . SHIFT      : POTENTIAL ENERGY SHIFT TO ENSURE THE TOTAL ENERGY       .
C .              IS ZERO                                                 .
C . SPRINGRF   : SPRING UNABALNCE FORCE VECTOR                           .
C . SPRINGEK   : SPRING STIFFNESS MATRIX                                 .
C . SPRINGEX   : ARRAY CONTAINING THE CURRENT COORDINATES OF THE         .
C .              SPRING ELEMENT                                          .
C . SPRINGEX0  : ARRAY CONTAINING THE ORIGINAL COORDINATES OF THE        .
C .              SPRING ELEMENT                                          .
```

```
C . STIFF     : STIFFNESS OF THE CREASE OR BOND                        .
C . U         : DISPLACEMENT INCREMENT                                 .
C . X         : ARRAY CONTAINING THE COORDINATES OF THE 3 NODES        .
C .             CONTRIBUTING AN ANGLE UNDER CONSIDERATION              .
C .                                                                    .
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
```

APPENDIX II

SAMPLE INPUT FILE TEMPLATES

The actual input files are too large to be included in the appendix. Hence, a format for both the element type has been shown below for readers' reference.

INPUT FILE TEMPLATE FOR MO

```
*HEAD
*NODE,NSET=ALL
**[+-- NODAL COORDINATES ---------------------------------]
.,.,.,.
*USER ELEMENT,NODES=9,TYPE=U1,COORDINATES=3,VARIABLES=15
1,2,3
*ELEMENT,TYPE=U1,ELSET=TEST
**[+-- CONNECTIVITY MATRIX FOR U1 ----------------------]
.,.,.,.
*UEL PROPERTY,ELSET=TEST
*USER ELEMENT,NODES=2,TYPE=U2,COORDINATES=3
1,2,3
*ELEMENT,TYPE=U2,ELSET=SPRING
**[+-- CONNECTIVITY MATRIX FOR U2 ----------------------]
.,.,.,.
*UEL PROPERTY,ELSET=SPRING
**[+-- NODE SETS ---------------------------------------]
*NSET,NSET=X0
.,.,.,.
*BOUNDARY
*STEP, INC=10000000000
**[+-- DISPLACEMENT BOUNDARY CONDITIONS -----------------]
STATIC
.,.,.,.
*CONTROLS,PARAMETERS=TIME INCREMENTATION
10000,10000,10000,10000,10000,10000
**[+-- LOADING -----------------------------------------]
*BOUNDARY
.,.,.,.
*NODE PRINT, NSET = ALL
U, RF
*END STEP
```

INPUT FILE TEMPLATE FOR RR

```
*HEAD
*NODE,NSET=ALL
**[+-- NODAL COORDINATES --------------------------------]
.,.,.,.
*USER ELEMENT,NODES=19,TYPE=U1,COORDINATES=3,VARIABLES=15
1,2,3
*ELEMENT,TYPE=U1,ELSET=TEST
**[+-- CONNECTIVITY MATRIX FOR U1 ----------------------]
.,.,.,.
*USER ELEMENT,NODES=19,TYPE=U2,COORDINATES=3,VARIABLES=15
1,2,3
*ELEMENT,TYPE=U2,ELSET=TEST
**[+-- CONNECTIVITY MATRIX FOR U2 ----------------------]
.,.,.,.
*USER ELEMENT,NODES=2,TYPE=U3,COORDINATES=3,VARIABLES=15
1,2,3
*ELEMENT,TYPE=U3,ELSET=TEST
**[+-- CONNECTIVITY MATRIX FOR U3 ----------------------]
.,.,.,.
*USER ELEMENT,NODES=2,TYPE=U4,COORDINATES=3,VARIABLES=15
1,2,3
*ELEMENT,TYPE=U4,ELSET=TEST
**[+-- CONNECTIVITY MATRIX FOR U4 ----------------------]
.,.,.,.
*UEL PROPERTY,ELSET=TEST
*USER ELEMENT,NODES=2,TYPE=U5,COORDINATES=3
1,2,3
*ELEMENT,TYPE=U5,ELSET=SPRING
**[+-- CONNECTIVITY MATRIX FOR U5 ----------------------]
.,.,.,.
*UEL PROPERTY,ELSET=SPRING
**[+-- NODE SETS ---------------------------------------]
*NSET,NSET=X0
.,.,.,.
*BOUNDARY
*STEP, INC=10000000000
**[+-- DISPLACEMENT BOUNDARY CONDITIONS -----------------]
STATIC
.,.,.,.
*CONTROLS,PARAMETERS=TIME INCREMENTATION
10000,10000,10000,10000,10000,10000
**[+-- LOADING -----------------------------------------]
*BOUNDARY
.,.,.,.
*NODE PRINT, NSET = ALL
U, RF
*END STEP
```

BIOGRAPHICAL SKETCH

Deepakshyam Krishnaraju is currently pursuing his Master's degree in the Mechanical Engineering Program at Arizona State University. In May 2014, he will graduate with a focus in computational Mechanics. Deepak is a member of a number of professional and student organizations including the Arizona State Chapter of SAE (Society of Automotive Engineers), and ASME (American Society of Mechanical Engineers). During spring 2013, Deepak participated in structural engineering internship with the Large Group, a nationally recognized AE firm, a structural engineering consulting firm concentrating in the refurbishment of damaged Aircraft parts. These experiences provided him with valuable experience and insight into the general principles of product design. Deepak is looking forward to pass the FE Exam this October and gain EIT status. In the short term, he looks forward to work as a Mechanical Engineer for a consulting or AE firm to gain knowledge and practical experience in the computational mechanics and Finite Element Analysis and to obtain his license as a Professional Engineer (PE). Outside of academics, Deepak is very involved in University activities. He spent couple of summers as a representative of ISA (Indian Students Association), a University service organization that assists International students from India on campus. Deepak enjoys outside activities including trail running, hiking, rock climbing and Class 4 and 5 white water rafting.